Building CTRnet Digital Library Services using Archive-It and LucidWorks Big Data Software

Kiran Chitturi

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Applications

Edward A. Fox, Chair
Steven D. Sheetz
Danfeng Yao

June 11, 2013
Blacksburg, Virginia

Keywords: Digital Library Services, CTRnet, Internet Archive, LucidWorks, Big Data, Crises, Archive-It
Copyright 2013, Kiran Chitturi
Building CTRnet Digital Library Services using Archive-It and LucidWorks Big Data Software

Kiran Chitturi

(ABSTRACT)

When a crisis occurs, information flows rapidly in the Web through social media, blogs, and news articles. The shared information captures the reactions, impacts, and responses from the government as well as the public. Later, researchers, scholars, students, and others seek information about earlier events, sometimes for cross-event analysis or comparison. There are very few integrated systems which try to collect and permanently archive the information about an event and provide access to the crisis information at the same time. In this thesis, we describe the CTRnet Digital Library and Archive which aims to permanently archive crisis event information by using Archive-It services and then provide access to the archived information by using LucidWorks Big Data software. Through the Big Data (LWBD) software, we take advantage of text extraction, clustering, similarity, annotation, and indexing services and build digital libraries with the generated metadata that will be helpful for the system stakeholders to locate information about an event.

Through this study, we collected data for 46 crises events using Archive-It. We built a CTRnet DL prototype and its services for the “Boston Marathon Bombing” collection by using the components of LucidWorks Big Data. Running LucidWorks Big Data on a 30 node Hadoop cluster accelerates the sub-workflows processing and also provides fault tolerant execution. LWBD sub-workflows, “ingest” and “extract”, processed the textual data present in the WARC files. Other sub-workflows “kmeans”, “simdoc”, and “annotate” helped in grouping the search-results, deleting the duplicates and providing metadata for additional facets in the CTRnet DL prototype, respectively.
Acknowledgments

I would have never been able to finish my thesis without the guidance of my committee members, help from friends, and support from my family.

I have been very fortunate to do my Masters thesis with Professor Edward A. Fox, who is not only my advisor and thesis committee chairman but also a mentor. I would have never survived my graduate program without the energy, encouragement, and inspiration from him. I am sincerely grateful for his excellent guidance, caring, patience and providing me with an excellent atmosphere all through my Masters.

My sincerest thanks must also go to committee member, Dr. Steven D. Sheetz, and to CTRnet Co-PIs Dr. Andrea L. Kavanugh and Dr. Donald J. Shoemaker. Their support and comments are much appreciated and have benefited this work as a result of their expertise and suggestions. Special thanks go to Dr. Danfeng Yao, who was willing to participate in my final defense committee at the last minute.

I would like to thank the Internet Archive and LucidWorks organizations for their constant support throughout my thesis. Special thanks go to Lori Donovan of the Internet Archive for helping me with the Archive-It software and Ivan Provalov of LucidWorks for his constant support during my work with LucidWorks Big Data Software.

I would also like to thank my friends and colleagues at the Digital Library Research Laboratory for their support, feedback, and friendship throughout my time here. Special thanks go to Seungwon Yang and Mohamed Magdy for helping me all through my thesis.

My journey here would not have been the same if I had not met my roommates Ananth, M.A., Nabeel, and Subbu. I am grateful for their friendship and support which made my journey so special and an unforgettable experience. Special thanks go to Krish who gave me best company since my starting day here.

My greatest thanks go out to my parents who helped me begin this, and all journeys. Your love and support throughout my education has helped me get where I am and accomplish all that I have. Additional thanks go to my brother for his encouragement through the years.

This work was supported in part by NSF grant no. IIS-0916733. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of NSF.
## Contents

1 Introduction

1.1 Motivation ............................................. 1

1.2 Problem Statement, Hypotheses, Research Questions .............. 3
  1.2.1 Problem Statement .................................. 3
  1.2.2 Hypotheses ......................................... 3
  1.2.3 Research Questions .................................. 4

1.3 Internet Archive, Archive-It and LucidWorks Big Data Software .. 4

1.4 Document Overview ...................................... 6

2 Background

2.1 Digital Libraries ........................................ 7
  2.1.1 Minimal DL .......................................... 7
  2.1.2 Complex Objects .................................... 8
  2.1.3 5S Approach ........................................ 9

2.2 Archive-It ................................................ 11
  2.2.1 Seed Types ......................................... 11
  2.2.2 Seed Frequency ..................................... 12
  2.2.3 Robots.txt .......................................... 13
  2.2.4 WARC Files ........................................ 13

2.3 LucidWorks Big Data Software ................................ 14
2.3.1 Architecture ........................................... 14
2.3.2 Apache Hadoop ........................................... 16
2.3.3 Apache Zookeeper ......................................... 18
2.3.4 Apache HBase ............................................ 19
2.3.5 Apache Mahout ............................................ 20
2.3.6 Apache Oozie .............................................. 21
2.3.7 Apache Kafka .............................................. 21
2.3.8 Apache Pig ................................................ 22
2.3.9 LucidWorks Search ........................................ 23

3 CTRnet DL Overview and Data Collection 25
3.1 CTRnet Digital Library Overview .............................. 25
3.2 Data Collection .............................................. 27
   3.2.1 Seed Collection ....................................... 27
   3.2.2 Crawling using Archive-It ................................ 32

4 WARC Processing with LucidWorks Big Data Software 40
4.1 Installation ................................................ 40
4.2 WARC Record .............................................. 41
4.3 ETL and sub-Workflows .................................... 45
   4.3.1 Ingest sub-workflow ................................ 47
   4.3.2 Extract sub-workflow ................................ 50
   4.3.3 Vectorize sub-workflow .............................. 55
   4.3.4 Annotations sub-workflow ............................ 55
   4.3.5 K-Means sub-workflow ............................... 57
   4.3.6 Similar Documents sub-workflow .................... 60
   4.3.7 Collocations sub-workflow ........................... 60
5  CTRnet Digital Library 65

5.1 5S Perspective ................................................. 65

5.1.1 Societies .................................................. 65

5.1.2 Streams ................................................... 67

5.1.3 Scenarios ................................................. 67

5.1.4 Structures ............................................... 67

5.1.5 Spaces ................................................... 68

5.2 Case Study: CTRnet DLA Prototype ............................. 68

5.2.1 Facets ................................................... 69

5.2.2 Clustered Documents ....................................... 70

5.2.3 LWBD Sub-workflow timings ............................... 70

5.2.4 Lessons Learned ......................................... 73

6  Conclusions and Future work 74

6.1 Conclusions ................................................. 74

6.2 Future Work ................................................. 75

Bibliography 77
List of Figures


3.1 Dataflow in CTRnet Digital Library ................................. 26
3.2 Dataflow during data collection .................................. 28

4.1 A complex object representing a WARC record .................. 42
4.2 Example of a ‘response’ record in a WARC file .................. 43
4.3 Example of a ‘request’ record in a WARC file .................... 44
4.4 Example of a ‘metadata’ record in a WARC file .................. 45
4.5 Example of a curl command executing a full ETL workflow for processing WARC files .............................................. 46
4.6 Full ETL workflow .................................................... 47
List of Tables

2.1 The specific components of LucidWorks Big Data as per version 1.1 .... 16

3.1 Top 7 URLs in the seed dump from first day of tweets collected about the
Boston Marathon Bombing .............................................. 30

3.2 Archive-It CTRnet collections [5] .................................. 34

3.3 CTRnet tweet collections [5] ........................................ 37

4.1 Services and co-locations in the System G CTRnet cluster ............. 41

4.2 Example: Data generated during ingest sub-workflow .................. 49

4.3 Example: Data generated during Tika sub-workflow .................... 52

4.4 Example: Date formats before and after conversion to ISO 8601 format . 55

4.5 Example: Data generated during K-Means sub-workflow ............... 60

4.6 Example: Data generated during Sim Doc sub-workflow ............... 62

5.1 WARC statistics for ‘Boston Marathon Bombings’ collection ........... 70

5.2 LWBD Sub-workflow timings for ‘Boston Marathon Bombings’ collection . 73
Chapter 1

Introduction

1.1 Motivation

When a crisis or disaster occurs, many users try to locate the most up-to-date information about that event. Later, researchers, scholars, students, and others seek information about earlier crisis events, sometimes for cross-event analysis or comparison. Yet, there is little systematic archiving anywhere of information about crisis events, except when national or state events are captured as part of government related Web archives (e.g., September 11, 2011 attacks), or when media companies (e.g., CSPAN) build archives of stories they cover. There are real time information systems like Google News, which show the latest news articles regarding a particular event, but they do not show the consolidated information long after that event.

The Crisis, Tragedy and Recovery network (CTRnet) [6] has emerged from a 4 year NSF IIS-0916733 supported grant, that aimed to collect and archive information about different types of crises and tragic events that have happened across the globe. The project is in
collaboration with the Internet Archive [7], that provides the Archive-It service which is used for data collection in this project. The significant crisis events are recognized and the seeds for Web crawls are given as input to the online Archive-It tool which completes the crawling based on the scope and depth specifications provided. The project archived over 40 different crises events and more than 10TB of data was collected about the crisis events. The information is present in the form of Web ARChive (WARC) [8] files that are preserved permanently within the Internet Archive servers. The content associated with archived URLs within the WARC files are given access to the public by the WayBack Machine [9].

These archives have a high change of falling into the “DUSTY ARCHIVE” [10] scenario, in which the Web archives are largely unused and gather digital dust. The data preserved in this scenario remains just that - a specimen preserved for uncertain future use [10]. To avoid this type of scenario, we need to push Web archives back onto the Web, so researchers and the public can extract the information they need for crises events from the archives. We need to bring the archives closer to the “NIRVANA” scenario [10] where they would be at once robust, standardized, and securely preserved while at the same time, open, flexible, widely used, and part of the standard research toolkit in academic disciplines. To achieve this, we need to use more powerful and effective tools for text search, information extraction, analysis, and visualization.

LucidWorks Big Data Software [11] is an application development platform that enables comprehensive Search, Discovery, and Analysis of organization’s content and user interactions and includes all of the necessary open source components pre-integrated and certified. We use the Big Data Software to extract text, cluster, annotate, and index the WARC files. Faceted Search service is provided by the CTRnet Digital Library, through which users can search and browse the information archived about the event. The software is installed in a cluster of 30 nodes in Virginia Tech’s System G supercomputer. The components present in
the software help us scale the processing for big data.

1.2 Problem Statement, Hypotheses, Research Questions

1.2.1 Problem Statement

An information explosion takes place over the Web for every crisis event through websites of various organizations (24 * 7 News Channels, Disaster Agencies) and Social Media (Twitter, Facebook, Reddit, etc.). All this varied information can be of great interest to researchers, government, and the public when trying to study about the event. It is challenging to permanently archive all this information and provide access to the stakeholders.

Today we have real-time information systems like Google News, which show the latest news articles regarding a particular event, but they do not show the consolidated information long after that event.

We propose building integrated digital library services developed from permanent archives for crisis events, that can be accessed by users to learn about the events.

1.2.2 Hypotheses

We hypothesize that

- Planning for the CTRnet digital library prototype can be effectively guided by treating it as a 5S case study.
• The LucidWorks Big Data Software, operating on a cluster of nodes using Virginia Tech’s System G, can efficiently and effectively support prototyping the key services needed for the CTRnet digital library.

1.2.3 Research Questions

Our work intends to address the following research questions:

• What data can be collected using, and then exported from, Archive-It, when building the local CTRnet DL prototype?

• How can the LucidWorks Big Data components be utilized in the CTRnet DL prototype?

• How can the LucidWorks Big Data sub-workflows be utilized in the CTRnet DL prototype?

1.3 Internet Archive, Archive-It and LucidWorks Big Data Software

Archive-It is an Internet Archive service that provides access services (browsing and searching) for the collections, that their partners crawled using Archive-It crawls. Archive-It allows its users to define three levels of metadata: 1) collection level metadata, 2) seed level metadata, and 3) document level metadata. The Dublin Core [12] schema is followed for the metadata definitions and customized metadata fields can also be created by the user.

While the data is crawled, the Archive-It user interface is updated with the seeds used in the
crawls and, in a period of 24 hours, the archived data is accessible from the user interface. The links are redirected to the WayBack machine, which stores the archived HTML pages. The metadata that is defined by the user is used for creating facets that help users navigate through the archived pages.

This approach is beneficial for users who do not have a large set of seeds for the crawl and are able to manually define metadata at both the seed level and document level. When the number of seeds that need to be crawled is high and the seeds vary every day for a different collection, it becomes hard to manually define metadata at the seed and document levels.

The thousands of webpages for every collection and the lack of rich metadata do not aid the user desiring better navigation through the pages. When the data volume dramatically increases for a collection, browse/search services are insufficient for a user to understand the data unless he already has a precise formulation of an information need. However, users generally do not know what to search for, in a news type service.

The seeds that are added by the curator are only shown in the Archive-It user interface but more pages are generated with in-depth crawls and these pages can only be ‘searchable’ for the end user. The depth crawls can produce pages that are related to the collection but these pages can also add to the set of non-relevant/spam pages which do not have any information about the event. We use keyword filters that are specific to an event and collection to remove this type of pages, often found within the WARC files.

To overcome these problems, we use the LucidWorks Big Data software to process the data to create additional metadata at the seed and document levels, and then use the clustering, annotation, and similar documents services to find similar stories, persons, locations, and organization names. We also use the similar documents to remove identically similar documents.
1.4 Document Overview

The rest of this thesis is organized as follows. In chapter 2, we discuss background information on i) Digital Libraries, ii) Archive-It, and iii) LucidWorks Big Data software and its components, which we use to build the services for the CTRnet digital library. In chapter 3, we describe the digital library and data collection process. In chapter 4, we discuss the use of LucidWorks Big Data software and its workflows. In chapter 5, we discuss the concepts and provide a 5S perspective of the CTRnet digital library by presenting our prototype as a case study. In chapter 6, we summarize support of stakeholders followed by recommending future work in the CTRnet or the follow-on IDEAL project.
Chapter 2

Background

2.1 Digital Libraries

Digital Libraries (DLs) are researched, developed, implemented, deployed, and used by millions of people in a wide variety of domains. They include advanced information systems that address the full information life cycle, facilitating asynchronous communication, across time and space, and enabling new methods for scholarly communication in our flat world. DLs can be immensely complex systems which allow information to be stored in an intelligent, usable, and easily retrievable fashion.

2.1.1 Minimal DL

In order to address the complexity of DLs, Goncalves et al. [13] proposed the 5S framework, where they define a “core” or “minimal” DL, i.e., the minimal set of components that make a DL, without which a system/application cannot be considered a DL. Figure 2.1 shows concepts in the metamodel for a minimal DL using the 5S framework. Arrows represent
dependencies, indicating that a concept is formally defined in terms of previously defined concepts that point to it.

2.1.2 Complex Objects

In order to reuse and aggregate different resources, Complex Objects (COs) [14] have been created, motivating solutions for integration and interoperability. Such objects are aggregation of different information combined together into a unique logical object.

A metadata specification describes a digital object. Some authors name the integration of
resources into a single digital object as Aggregation [15], a Component-Based Object [16], a Complex Object [15], or a Compound Object [17]. Categories for structuring digital objects include: atomistic, compound, and complex. The atomistic approach is when the user has a single file (whether made up from a single or multiple text files) from a preferred format. The compound approach is made up from multiple content files, which may have different formats. A complex object is described using a network of digital objects within a repository. [18]

2.1.3 5S Approach

According to the framework, the nature of DLs can be described using the 5S’s - Streams, Structures, Spaces, Scenarios, and Societies. Together these abstractions provide a formal foundation to define, relate, and unify concepts among others of digital objects, metadata, collections, and services - required to formalize and elucidate DLs. Through the 5S Approach, we can define digital libraries as complex systems that [19]:

1. help satisfy information needs of users (societies),
2. provide information services (scenarios),
3. organize information in usable ways (structures),
4. present information in usable ways (spaces), and
5. communicate information with users (streams).

STREAMS

Streams are sequences of arbitrary types (e.g., bits, characters, pixels, frames) and may be static or dynamic (such as audio and video). Streams describe properties of DL content such
as encoding and language for textual material or particular forms of multimedia data. [19]

**STRUCTURES**

A structure specifies the way in which parts of a whole are arranged or organized. In DLs, structures can represent hypertexts, taxonomies, system connections, user relationships, and containment - to cite a few. Books, for example, can be structured logically into chapters, sections, subsections, and paragraphs; or physically cover, pages, line groups (paragraphs), and lines. With the increase in heterogeneity of material continually being added to digital libraries, we find that much of this material is called “semi-structured” or “unstructured”. These terms refer to data that may have some structure, where the structure is not rigid, regular, explicit, or as complete a structure as is found in structured documents or traditional database management systems. [19]

**SPACES**

A space is a set of objects together with operations on those objects that obey certain constraints. Spaces define logical and presentational views of several DL components, and can be of type measurable, measure, probability, topological, metric, or vector space. [19]

**SCENARIOS**

A scenario is a sequence of events that also can have a number of parameters. Events represent changes in computational states; parameters represent specific variables defining a state and their respective values. Scenarios detail the behavior of DL services [19].
SOCITIES

A society is a “set of entities and the relationships between them” and can include both human users of a system as well as automatic software entities which have a certain role in system operation [19].

2.2 Archive-It

Archive-It is a subscription based Web archiving service that helps partner organizations harvest, build, and manage born digital collections. Through the Web application, Archive-It partners can collect, catalog, and manage the collections of archived content with 24/7 access and full text search available for their use. The partner base has steadily expanded since its launch, with 238 partners in forty-six U.S. States and fifteen countries, as of January 2013. Archive-It provides the user interface that lets the patrons create collections, add seeds, define the scope of seeds, add collection and document level metadata, and configure and run the crawls. The Heritrix open source software is used for crawling in the Archive-It service [20].

2.2.1 Seed Types

The seed URLs that are added to a collection can be configured as one of three seed types:

- Default
- RSS/News feed
- Crawl one page only
The seed type ‘Default’ will capture all links that are in scope.

The seed type ‘Crawl One Page Only’ will mean that only the seed URL page will be captured (i.e., no links to other pages will be captured). Any embedded images or other files (such as CSS or Javascript files) that are necessary in order to accurately render the page also will be captured.

The seed type ‘RSS/News feed’ is used when one wishes to crawl the RSS/News feed for that seed URL. The crawler will go to each URL that is listed in the feed and will crawl just that one URL (i.e., each URL in the feed will be crawled as ‘one page only’). This feature is used while crawling news sites like Google News.

### 2.2.2 Seed Frequency

Seed frequency defines the frequency for the crawls for the particular seed.

The crawl frequencies that are available with Archive-It are:

- Twice Daily
- Daily
- Weekly
- Monthly
- Bi-monthly
- Quarterly
- Semiannual
- One-Time
2.2.3 Robots.txt

Using Archive-It, we can crawl sites that disallow crawling by using the robots exclusion protocol. The robots exclusion protocol is a way for a webmaster to direct a Web crawler not to crawl all or specified parts of their website. By default, the Archive-It crawler honors and respects all robots.txt exclusion requests. On a case by case basis partners can set up rules to ignore robots.txt blocks for specific sites.

2.2.4 WARC Files

Collections are defined using Archive-It and a set of seeds and the data crawled from seeds is associated with that particular collection. Collections also can be seen as a set of WARC files.

The WARC (Web ARChive) format specifies a method for combining multiple digital resources into an aggregate archival file together with related information. The WARC format is a revision of the Internet Archive’s ARC format that has traditionally been used to store “Web crawls” as sequences of content blocks harvested from the World Wide Web. The WARC format generalizes the older format to better support the harvesting, access, and exchange needs of archiving organizations. Besides the primary content currently recorded, the revision accommodates related secondary content, such as assigned metadata, abbreviated duplicate detection events, and later-date transformations [21].

The metadata format of each document inside a WARC-file is defined as per the IIPC specification [21].
2.3 LucidWorks Big Data Software

LucidWorks Big Data (LWBD) is an integrated platform for enhancing data-driven decisions by leveraging tools for search, discovery, and analysis of massive data or content sets. Built on leading open source components such as Apache Solr [22], Apache Hadoop [23], Apache Mahout [24] and others, in addition to the power of LucidWorks Search [25], it brings together previously disparate tools into a single distribution with a unified, secure REST API for application integration and administration. This open core approach allows application developers the ability to focus on integration, while giving Data Scientists, Content Architects, Search Developers, and Big Data Engineers the ability to plug in scalable algorithms customized to meet business goals without having to wire together the low-level Hadoop ecosystem pieces time and again. Moreover, LucidWorks Big Data (LWBD) also provides the tools that developers need to manage and monitor a large scale, distributed cluster [26].

2.3.1 Architecture

As you might expect with a system that integrates several technologies, there are a variety of needs in terms of bringing the system together. Figure 2.3 illustrates how a number of these components work together.

The core technologies are LucidWorks Search (built on Lucene and Solr) and Apache Hadoop. These systems work together to process large amounts of data, using the features of Solr and Hadoop to bulk index massive amounts of content across a clustered environment. This allows the system to both serve content to end-users, analyze their search behavior, and provide the scalability to grow indefinitely [26].

The specific components of LucidWorks Big Data are listed in Table 2.1, with the version
### Table 2.1: The specific components of LucidWorks Big Data as per version 1.1

<table>
<thead>
<tr>
<th><strong>Product</strong></th>
<th><strong>Description</strong></th>
<th><strong>Version</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache Hadoop</td>
<td>Provides distributed storage and general purpose distributed computation</td>
<td>1.0.4</td>
</tr>
<tr>
<td>Apache HBase</td>
<td>Provides distributed storage for fast lookups based on Hadoop. Used to store metrics, user info, and history data.</td>
<td>0.94.1</td>
</tr>
<tr>
<td>Apache Hive</td>
<td>Provides data analysis</td>
<td>0.9.0</td>
</tr>
<tr>
<td>Apache Mahout</td>
<td>Provides scalable machine learning</td>
<td>0.8-SNAPSHOT</td>
</tr>
<tr>
<td>Apache Kafka</td>
<td>Provides a distributed publish-subscribe mechanism for real-time distributed data sharing, and for aggregating logs into HDFS.</td>
<td>0.7.1</td>
</tr>
<tr>
<td>Apache Oozie</td>
<td>Provides distributed workflow coordination</td>
<td>3.2.0-SNAPSHOT for compatibility with Hadoop 1.0.4</td>
</tr>
<tr>
<td>Apache Pig</td>
<td>Provides scripting language for manipulating large data sets for analytics and ETL</td>
<td>0.10.0</td>
</tr>
<tr>
<td>Apache Zookeeper</td>
<td>Provides distributed machine synchronization and configuration</td>
<td>3.4.3</td>
</tr>
<tr>
<td>LucidWorks Search</td>
<td>Provides search and discovery capabilities, plus connectors to common data sources</td>
<td>2.5, including Solr 4.0 and some patches for Kafka Integration</td>
</tr>
</tbody>
</table>

numbers used for each component:

### 2.3.2 Apache Hadoop

Apache Hadoop is a software project of the Apache Software Foundation. Apache Hadoop is a framework for running applications on large clusters built of commodity hardware. The Hadoop framework transparently provides applications both reliability and data motion. Hadoop implements a computational paradigm named MapReduce [27], where the application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster. In addition, it provides a distributed file system
(HDFS) [28] that stores data on the compute nodes, providing very high aggregate bandwidth across the cluster. Both MapReduce and the HDFS are designed so that node failures are automatically handled by the framework.

**Hadoop DFS**

Hadoop’s Distributed File System is designed to reliably store very large files across machines in a large cluster. It is inspired by the Google File System [29]. Hadoop DFS stores each file as a sequence of blocks; all blocks in a file except the last block are the same size. Blocks belonging to a file are replicated for fault tolerance. The block size and replication factor are configurable per file. Files in HDFS are “write once” and have strictly one writer at any time [30].

**Architecture**

An HDFS installation consists of a single Namenode, a master server that manages the filesystem namespace and regulates access to files by clients. In addition, there are a number of Datanodes, one per node in the cluster, which manage storage attached to the nodes that they run on. The Namenode makes filesystem namespace operations like opening, closing, renaming, etc. of files and directories available via an RPC interface. It also determines the mapping of blocks to Datanodes. The Datanodes are responsible for serving read and write requests from filesystem clients; they also perform block creation, deletion, and replication upon instruction from the Namenode [30].
Map/Reduce

Programming model and execution framework

Map/Reduce is a programming paradigm that expresses a large distributed computation as a sequence of distributed operations on data sets of key/value pairs. The Hadoop Map/Reduce framework harnesses a cluster of machines and executes user defined Map/Reduce jobs across the nodes in the cluster. A Map/Reduce computation has two phases, a map phase and a reduce phase. The input to the computation is a data set of key/value pairs.

Architecture

The Hadoop Map/Reduce framework has a master/slave architecture. It has a single master server or jobtracker and several slave servers or tasktrackers, one per node in the cluster. The jobtracker is the point of interaction between users and the framework. Users submit map/reduce jobs to the jobtracker, which puts them in a queue of pending jobs and executes them on a first-come/first-serve basis. The jobtracker manages the assignment of map and reduce tasks to the tasktrackers. The tasktrackers execute tasks upon instruction from the jobtracker and also handle data motion between the map and reduce phases [30].

2.3.3 Apache Zookeeper

Apache ZooKeeper [31] is a software project of the Apache Software Foundation. ZooKeeper allows distributed processes to coordinate with each other through a shared hierarchical name space of data registers (called znodes), much like a file system. Unlike normal file systems ZooKeeper provides its clients with high throughput, low latency, high availability, and strictly ordered access to the znodes [32].
Clients only connect to a single ZooKeeper server. The client maintains a TCP connection through which it sends requests, gets responses, gets watch events, and sends heartbeats. If the TCP connection to the server breaks, the client will connect to a different server. When a client first connects to the ZooKeeper service, the first ZooKeeper server will set up a session for the client. If the client needs to connect to another server, this session will get reestablished with the new server [32].

Zookeeper is used by HBase, Solr, and other components in the LucidWorks Big Data software.

2.3.4 Apache HBase

Apache HBase [33] is a software project of the Apache Software Foundation. HBase is a open source, non-relational, distributed database modeled after Google’s BigTable [34]. It is developed as part of the Apache Hadoop project and runs on top of HDFS, providing BigTable-like capabilities for Hadoop.

HBase is a NoSQL database: the Hadoop database. It’s often described as a sparse, distributed, persistent, multidimensional sorted map, which is indexed by row key, column key, and timestamp. Fundamentally, HBase is a platform for storing and retrieving data with random access, meaning you can write data as you like and read it back again as you need it [35].

Architecture

HBase is designed to run on a cluster of computers instead of a single computer. It is built on top of HDFS and provides fast record lookups (and updates) for large tables. HBase internally puts the data in indexed “StoreFiles” that exist on HDFS for high-speed lookups.
Regions and RegionServers

Tables in HBase comprise rows and columns, albeit with a different kind of schema which can scale to billions of rows and millions of columns. The size of each table can run into terabytes and sometimes even petabytes. It’s clear at that scale that an entire table can’t be hosted on a single machine. Instead, tables are split into smaller chunks that are distributed across multiple servers. These smaller chunks are called regions. Servers that host regions are called RegionServers.

RegionServers are typically collocated with HDFS DataNodes on the same physical hardware, although that’s not a requirement. The only requirement is that RegionServers should be able to access HDFS. They’re essentially clients and store/access data on HDFS. The master processes the distribution of regions among RegionServers, and each RegionServer typically hosts multiple regions.

2.3.5 Apache Mahout

Apache Mahout [24] is an open source machine learning library project of the Apache Software Foundation. The algorithms it implements fall under the broad umbrella of machine learning or collective intelligence. It is scalable to reasonably large data sets and the core algorithms for clustering, classification, and batch based collaborative filtering are implemented on top of Apache Hadoop using the map/reduce paradigm.

Currently Mahout supports mainly four use cases: Recommendation mining takes user’s behavior and from that tries to find items users might like. Clustering takes, e.g., text documents, and aggregates them into groups of topically related documents. Classification learns from existing categorized documents what documents of a specific category look like and is able to assign unlabeled documents to the (hopefully) correct category. Frequent item
set mining takes a set of item groups (terms in a query session, shopping cart content) and identifies which individual items usually appear together [24].

### 2.3.6 Apache Oozie

Apache Oozie [36] is a software project of the Apache Software Foundation. Oozie is a workflow scheduler system to manage Apache Hadoop jobs. The workflow jobs are Directed Acyclical Graphs (DAGs) of actions. Oozie is integrated with the rest of the Hadoop stack supporting several types of Hadoop jobs out of the box (such as Java MapReduce, Streaming MapReduce, Pig, Hive, Sqoop, and Distcp) as well as system specific jobs (such as Java programs and shell scripts).

Oozie provides a user interface, that lets users check the status of the jobs that are submitted.

### 2.3.7 Apache Kafka

Apache Kafka [37] is a software project of the Apache Software Foundation. Apache Kafka is a distributed publish-subscribe messaging system. It is designed to support:

- Persistent messaging with O(1) disk structures that provide constant time performance even with many terabytes of stored messages.

- High-throughput: even with very modest hardware Kafka can support hundreds of thousands of messages per second.

- Explicit support for partitioning messages over Kafka servers and distributing consumption over a cluster of consumer machines while maintaining per-partition ordering semantics.
Support for parallel data load into Hadoop.

2.3.8 Apache Pig

Apache Pig is a software project of the Apache Software Foundation. Pig provides an engine for executing data flows in parallel on Hadoop. It includes a language, Pig Latin, for expressing these data flows. Pig Latin includes operators for users to develop their own functions for reading, processing, and writing data.

Pig runs on Hadoop. It makes use of both the Hadoop Distributed File System, HDFS, and Hadoop’s processing system, MapReduce.

By default, Pig reads input files from HDFS, uses HDFS to store intermediate data between MapReduce jobs, and writes its output to HDFS. It also can read input from and write output to sources other than HDFS, like HBase, etc.

Pig Latin, a Parallel Dataflow Language

Pig Latin [38] is a dataflow language. This means it allows users to describe how data from one or more inputs should be read, processed, and then stored to one or more outputs in parallel. These data flows can be simple linear flows like word count and also can be complex workflows that include points where multiple inputs are joined, and where data is split into multiple streams to be processed by different operators.

Pig compiles these dataflow programs, which are written in Pig Latin, into sets of Hadoop MapReduce jobs, and coordinates their execution.

Pig got its start from Yahoo in 2006 [38] and 40% of all Hadoop jobs in Yahoo are run with Pig.
2.3.9 LucidWorks Search

LucidWorks Search is the search solution development platform built on the power of Apache Solr/Lucene, developed by the enterprise search experts at LucidWorks. LucidWorks Search leverages the disruptive innovation of the leading open source search technology, to deliver unmatched scalability to billions of documents, with sub-second query and faceting response time. By building and extending the scalable power of Solr open source with vital new features, the search experts at LucidWorks have created an integrated platform that simplifies and empowers predictable, reliable search application development [39].

LucidWorks Search offers a Web-based user interface that provides easy ways to accomplish common tasks and also, a Search ReST API which enables users to accomplish tasks programmatically using HTTP calls. It is open core, meaning the user has the flexibility to modify the Apache Solr open source component.

Apache Solr

Apache Solr is a software project of the Apache Software Foundation. Solr is the popular, blazingly fast open source enterprise search platform from the Apache Lucene project. Its major features include powerful full-text search, hit highlighting, faceted search, near real-time indexing, dynamic clustering, database integration, rich document (e.g., Word, PDF) handling, and geospatial search. Solr is highly reliable, scalable and fault tolerant, providing distributed indexing, replication and load-balanced querying, automated failover and recovery, centralized configuration, and more. Solr powers the search and navigation features of many of the world’s largest internet sites [22].

Solr is written in Java and runs as a standalone full-text search server within a servlet container such as Jetty. Solr uses the Lucene Java search library at its core for full-text
indexing and search, and has REST-like HTTP/XML and JSON APIs that make it easy to use from virtually any programming language. Solr’s powerful external configuration allows it to be tailored to almost any type of application without Java coding, and it has an extensive plugin architecture when more advanced customization is required [22].

From version 4.0 release, Solr offers two modes of operation. One is a single node server. The second one is a distributed SolrCloud.

**SolrCloud**

SolrCloud [40] is the name of a set of distributed capabilities in Solr. Specifying your configuration to enable these capabilities will let you set up a highly available, fault tolerant cluster of Solr servers. SolrCloud can be used when one wants highly scalable, fault tolerant, and distributed indexing and search capabilities.

SolrCloud supports the following features:

- Central configuration for the entire cluster
- Automatic load balancing and fail-over for queries
- ZooKeeper integration for cluster coordination and configuration.

SolrCloud is like a flexible search, except that there is no master node to allocate nodes, shards, and replicas. Instead, Solr uses ZooKeeper to manage these locations, depending on configuration files and schemas. Documents can be sent to any server and ZooKeeper will figure it out.
Chapter 3

CTRnet DL Overview and Data Collection

3.1 CTRnet Digital Library Overview

The CTRnet Digital Library (CTRnet DL) focuses on collecting information related to crisis events from the Web and making the information accessible to the system stakeholders through the user interface.

The three important phases in the construction of a Digital Library are:

1. Data Collection
2. Data Processing
3. User Interface

Figure 3.1 shows the data flow inside the CTRnet Digital Library.
Figure 3.1: Dataflow in CTRnet Digital Library
Archive-It is used for data collection and LucidWorks Big Data Software is used for data processing. We used Python’s lightweight Web framework (CherryPy \cite{41} + Jinja2 \cite{42}) for the User Interface.

### 3.2 Data Collection

During the Data Collection, we gather information about crisis events we are interested in and we use the Internet Archive Archive-It tool for crawling. The seeds for crawling come from two different sources: one is from tweets and the other is manual input by the public. Figure 3.2 shows the data flow in our ‘Data Collection’ process.

#### 3.2.1 Seed Collection

To crawl information about an event, we need to give seeds (URLs) as input to the crawler.

**Tweets DataSource**

Tweets are one of the most popular ways of spreading information among one’s followers and sharing with the Web through Twitter. Though tweet length is limited to 140 characters, the URLs tweeted can serve as high quality information or spam \cite{43}. We use the tweets as a source of information by extracting the seeds (URLs) that are shared with tweets.

**Tweet Collection**

We collect tweets using the Twitter Streaming and Searching Application Programming Interface (API), and store them inside the MySQL database in our local server. We collect
Figure 3.2: Dataflow during data collection
keywords and hashtags corresponding to the crisis event we are studying and then we form a tweet collection by using the keywords/hashtagsthrough the API.

**Tweet Analysis**

Python scripts are used to select the tweets from the database and the URLs are extracted from tweets using the twitter-text-python [44] library. The extracted URLs are short URLs that are expanded by using the Python requests library.

Though a short URL might be distinct when shared with the tweet, there are lots of duplicates present among the expanded URLs. The frequency of the extracted seeds is also calculated, and dumped into a text file along with the seeds.

We found a total of 9394 unique URLs from the first 150,000 tweets we collected about the Boston Marathon Bombing.

We have observed different types of resources being shared that are of interest to the users during an event. These resources range from static webpages to a live TV stream from a news channel. The streaming channels like news streams, police scanners, and radio broadcasts prove to be challenging to capture and archive in real time.

Table 1 shows the top 7 URLs out of 150,000 tweets collected on first day.

**Issues with Shared URLs**

Sometimes even though the URLs look distinct, they will be pointing to the same resource. This happens due to the additional parameters passed to the URL like ‘utm_source’, ‘utm_medium’, ‘utm_campaign’, etc. These parameters keep track of the source and medium which is later used in analytics by the host websites.
Table 3.1: Top 7 URLs in the seed dump from first day of tweets collected about the Boston Marathon Bombing

<table>
<thead>
<tr>
<th>Seed</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.youtube.com/watch?v=046MuD1pYJg&amp;feature=youtu.be">http://www.youtube.com/watch?v=046MuD1pYJg&amp;feature=youtu.be</a></td>
<td>923</td>
</tr>
<tr>
<td><a href="https://twitter.com/AitorOrtiz/status/324245175977267200/photo/1">https://twitter.com/AitorOrtiz/status/324245175977267200/photo/1</a></td>
<td>456</td>
</tr>
<tr>
<td><a href="http://bigstory.ap.org/article/trauma-doctor-most-boston-blast-injuries-legs">http://bigstory.ap.org/article/trauma-doctor-most-boston-blast-injuries-legs</a></td>
<td>441</td>
</tr>
</tbody>
</table>
For example, a news article by CNN referred through the CNN website is identified by the URL:


The same resource when referred from different websites, or through different sources, is identified in the following URLs:


The crawler treats all these URLs as different pages, and due to this reason, the crawled data contains many duplicate HTML files. It would be best to have a filtering mechanism in the future to prevent duplicates among the seeds rather than doing deduplication after the data is crawled or handle this in a focused crawler.

**User Seeds**

For several crisis collections, we used user input for seeds in addition to the seeds from tweets. Our partner, Internet Archive, has set up a publicly open Google Doc for several events through which users/subject experts can add seeds of interest.
3.2.2 Crawling using Archive-It

We use Archive-It to crawl the seeds that are collected through tweets and user input.

Adding Seeds

We use the Archive-It Admin screen to create a collection for an event and to add seeds for the particular collection. Currently, addition of seeds is only supported through the user interface screen and so needs to be done manually.

The seeds are selected through the Seed Collection process described in Section 3.1.1.

Seeds are filtered by using a threshold on the frequency of the seeds. For example, we select seeds that have a frequency of more than 2. The seeds from user input that are collected through a Google Doc are directly added to the seed list of the collection.

We added 180,982 seeds for the Hurricane Sandy collection and 9394 seeds for the Boston Marathon Bombing.

Collection, Seeds, Document Level Metadata

Archive-It allows all its partners to enter metadata on these three levels.

Dublin Core metadata fields are provided for defining metadata, and partners also can enter customized metadata fields. The collection level metadata is available for harvesting via OAI-PMH feeds.

Since the seeds are high in number, it is hard to define metadata manually for each of these seeds. We generate seed level and document level metadata during data processing and use this metadata for providing a faceted search.
Scope and Seed type definitions

All the seeds need to be provided with two main configurations. They are:

1) Crawl Frequency
2) Seed Type

The seeds that are extracted from tweets are listed with a crawl frequency of One-Time and seed type ‘Crawl one page only’ since these seeds represent static news articles.

The user-listed seeds are sometimes specified with a crawl frequency and seed type: otherwise we add an appropriate values for the seeds based on the resource type.

WARC Files

Archive-It uses the Heritrix crawler in the backend to crawl the seeds. Heritrix produces WARC files as output resulting from the crawls. These WARC files are available to download for partners via an admin user interface. We download these WARC files for all the collections to our servers and these files are used as input to LucidWorks Big Data Software. Sizes of the WARC files range between 10MB-2GB.

Table 3.2 shows all the collections that are archived in Archive-It between 2008 and 2013 by the CTRnet project.

Table 3.3 shows all the twitter collections that are archived between 2012 and 2013 by the CTRnet project.
Table 3.2: Archive-It CTRnet collections statistics. Collections can be browsed at [5]

<table>
<thead>
<tr>
<th>Name</th>
<th>Creation Date</th>
<th>No. of URLs</th>
<th>Seeds</th>
<th>Archive Size (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alabama University Shooting</td>
<td>Feb. 24, 2010</td>
<td>38K</td>
<td>116</td>
<td>4.1</td>
</tr>
<tr>
<td>April 16 Archive</td>
<td>May 21, 2007</td>
<td>4M</td>
<td>88</td>
<td>288</td>
</tr>
<tr>
<td>Blasts in Boston Marathon: Twitter and RSS feeds</td>
<td>June 26, 2013</td>
<td>6M</td>
<td>9394</td>
<td>519</td>
</tr>
<tr>
<td>Brazil Nightclub Fire</td>
<td>Feb. 1, 2013</td>
<td>2M</td>
<td>250</td>
<td>95</td>
</tr>
<tr>
<td>Brazilian School Shooting</td>
<td>Apr. 9, 2011</td>
<td>38K</td>
<td>650</td>
<td>0.05</td>
</tr>
<tr>
<td>Center for Research on the Epidemiology of Disasters (CRED) archive</td>
<td>Jan. 3, 2012</td>
<td>97K</td>
<td>1</td>
<td>1.1</td>
</tr>
<tr>
<td>Chile Earthquake</td>
<td>Oct. 5, 2009</td>
<td>2.5k</td>
<td>19</td>
<td>0.0051</td>
</tr>
<tr>
<td>China Earthquake</td>
<td>April 19, 2013</td>
<td>0.2M</td>
<td>291</td>
<td>11</td>
</tr>
<tr>
<td>CTRnet-Emergency Preparedness Information</td>
<td>Sep. 3, 2011</td>
<td>0.95M</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>Cyclone Yasi</td>
<td>Feb. 4, 2011</td>
<td>41K</td>
<td>319</td>
<td>1.9</td>
</tr>
<tr>
<td>East River Helicopter Crash</td>
<td>Oct. 5, 2011</td>
<td>5k</td>
<td>64</td>
<td>0.131</td>
</tr>
<tr>
<td>Encephalitis (India)</td>
<td>Oct. 12, 2011</td>
<td>2.6K</td>
<td>59</td>
<td>0.057</td>
</tr>
</tbody>
</table>

Continued on next page
Table 3.2 – continued from previous page

<table>
<thead>
<tr>
<th>Name</th>
<th>Creation Date</th>
<th>No. of URLs</th>
<th>Seeds</th>
<th>Archive Size (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Food Crisis</td>
<td>Oct. 19, 2011</td>
<td>6 M</td>
<td>130</td>
<td>717</td>
</tr>
<tr>
<td>Guatemala Earthquake</td>
<td>Nov. 8, 2012</td>
<td>2M</td>
<td>97</td>
<td>522</td>
</tr>
<tr>
<td>Gunman Reported at Virginia Tech</td>
<td>Aug. 7, 2011</td>
<td>18K</td>
<td>911</td>
<td>0.352</td>
</tr>
<tr>
<td>Haiti Earthquake Anniversary</td>
<td>Jan. 13, 2011</td>
<td>79K</td>
<td>70</td>
<td>2.4</td>
</tr>
<tr>
<td>Hurricane Irene</td>
<td>Sep. 1, 2011</td>
<td>79K</td>
<td>70</td>
<td>2.4</td>
</tr>
<tr>
<td>Hurricane Isaac 2012</td>
<td>Sep. 2, 2012</td>
<td>252K</td>
<td>2,492</td>
<td>5.5</td>
</tr>
<tr>
<td>Hurricane Sandy</td>
<td>Oct. 26, 2012</td>
<td>13M</td>
<td>180982</td>
<td>888</td>
</tr>
<tr>
<td>Indonesia Plane Crash</td>
<td>Oct. 2, 2011</td>
<td>13K</td>
<td>350</td>
<td>0.332</td>
</tr>
<tr>
<td>Indonesian Volcanic Eruption, Tsunami, Earthquake in 2010</td>
<td>Oct. 30, 2010</td>
<td>2M</td>
<td>1,121</td>
<td>148</td>
</tr>
<tr>
<td>Indonesian Volcanic Eruption, Tsunami, Earthquake in 2010 (Part 2)</td>
<td>Nov. 8, 2010</td>
<td>78K</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Japan Earthquake</td>
<td>Mar. 11, 2011</td>
<td>128M</td>
<td>19857</td>
<td>6963.2</td>
</tr>
<tr>
<td>International School Shootings</td>
<td>Apr. 26, 2010</td>
<td>1K</td>
<td>41</td>
<td>0.03</td>
</tr>
<tr>
<td>MidWest Snowstorms</td>
<td>Feb. 4, 201</td>
<td>275K</td>
<td>415</td>
<td>14</td>
</tr>
<tr>
<td>Nepal Plane Crash</td>
<td>Sep. 27, 2011</td>
<td>23K</td>
<td>629</td>
<td>0.76</td>
</tr>
<tr>
<td>Nevada air race crash</td>
<td>Sep. 17, 2011</td>
<td>11.3K</td>
<td>64</td>
<td>0.311</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Name</th>
<th>Creation Date</th>
<th>No. of URLs</th>
<th>Seeds</th>
<th>Archive Size (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Zealand Earthquake</td>
<td>Feb. 28, 2011</td>
<td>3.5K</td>
<td>44</td>
<td>0.109</td>
</tr>
<tr>
<td>Northern Illinois University Shooting</td>
<td>Feb. 15, 2008</td>
<td>631K</td>
<td>24</td>
<td>46</td>
</tr>
<tr>
<td>Norway Shooting</td>
<td>Jul. 24, 2011</td>
<td>80K</td>
<td>4,250</td>
<td>2</td>
</tr>
<tr>
<td>Pakistan floods</td>
<td>Sep. 15, 2011</td>
<td>106K</td>
<td>655</td>
<td>2.2</td>
</tr>
<tr>
<td>Philippines Floods</td>
<td>Sep. 28, 2011</td>
<td>22K</td>
<td>1,145</td>
<td>1.1</td>
</tr>
<tr>
<td>Russia Plane Crash</td>
<td>Sep. 7, 2011</td>
<td>311K</td>
<td>104</td>
<td>19</td>
</tr>
<tr>
<td>Sikkim Earthquake</td>
<td>Sep. 19, 2011</td>
<td>19K</td>
<td>171</td>
<td>0.472</td>
</tr>
<tr>
<td>Somalia Bomb Blast</td>
<td>Oct. 19, 2011</td>
<td>4K</td>
<td>61</td>
<td>0.177</td>
</tr>
<tr>
<td>South-Eastern US Storms</td>
<td>Apr. 28, 2011</td>
<td>22K</td>
<td>402</td>
<td>0.642</td>
</tr>
<tr>
<td>Texas fertilizer plant explosion</td>
<td>Apr. 18, 2013</td>
<td>2M</td>
<td>271</td>
<td>142</td>
</tr>
<tr>
<td>Texas Wild fire 2011</td>
<td>Sep. 7, 2011</td>
<td>350K</td>
<td>2,330</td>
<td>34</td>
</tr>
<tr>
<td>Thai Floods</td>
<td>Oct. 5, 2011</td>
<td>8.9K</td>
<td>64</td>
<td>0.166</td>
</tr>
<tr>
<td>Tucson Shooting Anniversary 2012</td>
<td>Jan. 11, 2012</td>
<td>7K</td>
<td>72</td>
<td>0.450</td>
</tr>
<tr>
<td>Tucson Shootings</td>
<td>Jan. 9, 2011</td>
<td>251K</td>
<td>1,996</td>
<td>8.1</td>
</tr>
<tr>
<td>Turkey Earthquake</td>
<td>Oct. 24, 2011</td>
<td>1M</td>
<td>686</td>
<td>35</td>
</tr>
<tr>
<td>Virginia Earthquake</td>
<td>Sep. 4, 2011</td>
<td>45K</td>
<td>1,544</td>
<td>1.4</td>
</tr>
<tr>
<td>Virginia Tech Global Disasters Collection</td>
<td>Apr. 19, 2011</td>
<td>12K</td>
<td>469</td>
<td>0.275</td>
</tr>
<tr>
<td>Virginia Tech Shootings</td>
<td>Dec. 8, 2011</td>
<td>1.5M</td>
<td>348</td>
<td>58</td>
</tr>
</tbody>
</table>
Table 3.2 – continued from previous page

<table>
<thead>
<tr>
<th>Name</th>
<th>Creation Date</th>
<th>No. of URLs</th>
<th>Seeds</th>
<th>Archive Size (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virginia Tech April 16 Shootings Remembrance</td>
<td>Apr. 16, 2010</td>
<td>14K</td>
<td>241</td>
<td>2.5</td>
</tr>
<tr>
<td>Youngstown Shootings</td>
<td>Feb. 8, 2011</td>
<td>34K</td>
<td>85</td>
<td>2.6</td>
</tr>
<tr>
<td>Zanzibar ferry disaster 2011</td>
<td>Sep. 11, 2011</td>
<td>54K</td>
<td>412</td>
<td>1.9</td>
</tr>
<tr>
<td><strong>Total Size</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>10,634,065</strong></td>
</tr>
</tbody>
</table>

Table 3.3: CTRnet Tweet Collections

<table>
<thead>
<tr>
<th>Keyword/Hashtag</th>
<th>Count</th>
<th>Time Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>@craigatFEMA</td>
<td>6301</td>
<td>05 Nov. 2012</td>
</tr>
<tr>
<td>@FEMA</td>
<td>63170</td>
<td>03 Nov. 2012</td>
</tr>
<tr>
<td>@NOAA</td>
<td>33361</td>
<td>03 Nov. 2012</td>
</tr>
<tr>
<td>@ReadydotGov</td>
<td>4391</td>
<td>05 Nov. 2012</td>
</tr>
<tr>
<td>@RedCross</td>
<td>248329</td>
<td>05 Nov. 2012</td>
</tr>
<tr>
<td>@SalvationArmy</td>
<td>18962</td>
<td>05 Nov. 2012</td>
</tr>
<tr>
<td>@SalvationArmyUS</td>
<td>20950</td>
<td>05 Nov. 2012</td>
</tr>
<tr>
<td>#Andrea</td>
<td>12486</td>
<td>06 June 2013</td>
</tr>
<tr>
<td>#bahrain</td>
<td>11226273</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>#blacksburg</td>
<td>12001</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>#egypt</td>
<td>4142477</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>#iran</td>
<td>671382</td>
<td>11 Feb. 2013</td>
</tr>
<tr>
<td>#iranelection</td>
<td>1291</td>
<td>08 Jun. 2013</td>
</tr>
<tr>
<td>#Iranelections</td>
<td>961</td>
<td>27 May. 2013</td>
</tr>
<tr>
<td>#Isaac</td>
<td>33252</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>#jan25</td>
<td>517897</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>#libya</td>
<td>530809</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>#nrv</td>
<td>26357</td>
<td>07 Oct. 2012</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Keyword/Hashtag</th>
<th>Count</th>
<th>Time Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>#syria</td>
<td>7256987</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>#yemen</td>
<td>586073</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>747 crash</td>
<td>5349</td>
<td>01 May. 2013</td>
</tr>
<tr>
<td>bangladesh building collapse</td>
<td>105462</td>
<td>24 Apr. 2013</td>
</tr>
<tr>
<td>boston blast</td>
<td>73489</td>
<td>15 Apr. 2013</td>
</tr>
<tr>
<td>boston explosion</td>
<td>133486</td>
<td>15 Apr. 2013</td>
</tr>
<tr>
<td>boston shooters</td>
<td>4526</td>
<td>19 Apr. 2013</td>
</tr>
<tr>
<td>brazil nightclub fire</td>
<td>83380</td>
<td>27 Jan. 2013</td>
</tr>
<tr>
<td>china earthquake</td>
<td>53611</td>
<td>19 Apr. 2013</td>
</tr>
<tr>
<td>christiansburg mall shooting</td>
<td>467</td>
<td>15 Apr. 2013</td>
</tr>
<tr>
<td>connecticut school shooting</td>
<td>96154</td>
<td>14 Dec. 2012</td>
</tr>
<tr>
<td>earthquake</td>
<td>3637022</td>
<td>26 Oct. 2012</td>
</tr>
<tr>
<td>emergency management</td>
<td>72489</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>emergency mitigation</td>
<td>655</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>emergency recovery</td>
<td>9452</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>foursquare</td>
<td>31406253</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>guatemala earthquake</td>
<td>42165</td>
<td>07 Nov. 2012</td>
</tr>
<tr>
<td>gun control</td>
<td>1429125</td>
<td>19 Jan. 2013</td>
</tr>
<tr>
<td>gun violence</td>
<td>377948</td>
<td>18 Feb. 2013</td>
</tr>
<tr>
<td>heart attack</td>
<td>7293329</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>hurricane</td>
<td>6566263</td>
<td>26 Oct. 2012</td>
</tr>
<tr>
<td>hurricane isaac</td>
<td>8031</td>
<td>01 Nov. 2012</td>
</tr>
<tr>
<td>hurricane sandy</td>
<td>2798703</td>
<td>26 Oct. 2012</td>
</tr>
<tr>
<td>iran earthquake</td>
<td>85647</td>
<td>07 Oct. 2012</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Keyword/Hashtag</th>
<th>Count</th>
<th>Time Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>iran election</td>
<td>636</td>
<td>08 Jun. 2013</td>
</tr>
<tr>
<td>Israel</td>
<td>7626702</td>
<td>15 Nov. 2012</td>
</tr>
<tr>
<td>kentucky shooting</td>
<td>16740</td>
<td>16 Jan. 2013</td>
</tr>
<tr>
<td>northeast storm</td>
<td>64470</td>
<td>08 Feb. 2013</td>
</tr>
<tr>
<td>obesity</td>
<td>1183862</td>
<td>16 Nov. 2012</td>
</tr>
<tr>
<td>OccupyWallStreet</td>
<td>167017</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>oklahoma tornado</td>
<td>376447</td>
<td>20 May. 2013</td>
</tr>
<tr>
<td>ontario earthquake</td>
<td>373</td>
<td>20 May. 2013</td>
</tr>
<tr>
<td>oregon shooting</td>
<td>20200</td>
<td>12 Dec. 2012</td>
</tr>
<tr>
<td>pakistan earthquake</td>
<td>10787</td>
<td>18 Apr. 2013</td>
</tr>
<tr>
<td>philadelphia building collapse</td>
<td>7693</td>
<td>05 Jun. 2013</td>
</tr>
<tr>
<td>Quantico shooting</td>
<td>5181</td>
<td>22 Mar. 2013</td>
</tr>
<tr>
<td>santa cruz earthquake</td>
<td>20245</td>
<td>07 Feb. 2013</td>
</tr>
<tr>
<td>santa monical shooting</td>
<td>16792</td>
<td>07 Jun. 2013</td>
</tr>
<tr>
<td>solomon islands earthquake</td>
<td>16920</td>
<td>06 Feb. 2013</td>
</tr>
<tr>
<td>terrorism</td>
<td>1532227</td>
<td>26 Oct. 2012</td>
</tr>
<tr>
<td>texas fertilizer explosion</td>
<td>49522</td>
<td>18 Apr. 2013</td>
</tr>
<tr>
<td>tsunami</td>
<td>1584899</td>
<td>05 Feb. 2013</td>
</tr>
<tr>
<td>tucson shooting</td>
<td>12188</td>
<td>08 Jan. 2013</td>
</tr>
<tr>
<td>tunisia</td>
<td>432667</td>
<td>06 Feb. 2013</td>
</tr>
<tr>
<td>turkey syria</td>
<td>271883</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>turkish car bombing</td>
<td>287</td>
<td>13 May. 2013</td>
</tr>
<tr>
<td>typhoon</td>
<td>465335</td>
<td>05 Dec. 2012</td>
</tr>
<tr>
<td>virginia tech</td>
<td>280889</td>
<td>07 Oct. 2012</td>
</tr>
<tr>
<td>virginia tech remembrance</td>
<td>970</td>
<td>15 Apr. 2013</td>
</tr>
<tr>
<td><strong>Total Count</strong></td>
<td>101,187,518</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 4

WARC Processing with LucidWorks

Big Data Software

4.1 Installation

LucidWorks Big Data (LWBD) software is installed on a cluster of 30 nodes in Virginia Tech’s System G, its green supercomputer. Each node is configured with 8GB RAM and 320 GB hard disk space with the CentOs operating system.

Since the software is tested on the Ubuntu operating system and requires frequent installations/upgrades, we decided to use virtual machines on each node. All the virtual machines are configured with an internal static IP address. We used one virtual node as head node for installations and as a Chef server [45]. The virtual machine host-names are named from LucidN1-LucidN29.

In the LucidWorks Big Data software, HBase is used as a data source and a data sink for all the services.
Table 4.1: Services and co-locations in the System G CTRnet cluster

<table>
<thead>
<tr>
<th>Service</th>
<th>Number of Nodes</th>
<th>Hostnames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zookeeper, Kafka</td>
<td>3</td>
<td>LucidN1, LucidN2, LucidN3</td>
</tr>
<tr>
<td>Oozie</td>
<td>1</td>
<td>LucidN2</td>
</tr>
<tr>
<td>SDA</td>
<td>1</td>
<td>LucidN3</td>
</tr>
<tr>
<td>Secondary Namenode</td>
<td>1</td>
<td>LucidN2</td>
</tr>
<tr>
<td>Namenode, Jobtracker, HBase Master</td>
<td>1</td>
<td>LucidN1</td>
</tr>
<tr>
<td>Datanode, RegionServer</td>
<td>20</td>
<td>LucidN10 - LucidN29</td>
</tr>
<tr>
<td>SDA-Events</td>
<td>21</td>
<td>LucidN3, LucidN10 - LucidN29</td>
</tr>
<tr>
<td>LucidWorks/Solr core</td>
<td>6</td>
<td>LucidN4 - LucidN9</td>
</tr>
</tbody>
</table>

The Big Data cluster uses the Chef Server [45] tool for node discovery and centralized configuration. Using Chef, we co-located the different services on the 20 nodes as shown in Table 4.1.

### 4.2 WARC Record

Many resources are integrated into a WARC file as a Complex Object. According to Krafft et al., Complex Objects are single entities that are composed of multiple digital objects, each of which is an entity in and of itself [46]. Figure 4.1 shows the higher level structure representation of a WARC record.

Every WARC file has a record information object which is defined as “warcinfo” [47]. All the other objects in the WARC file can be distinguished as different URIs, with three types of digital objects associated with a URI. The three objects are distinguished by distinct values of ‘WARC-Type’ field in the metadata:
Figure 4.1: A complex object representing a WARC record

**Response**

A ‘response’ [48] record contains a complete scheme-specific response, including network protocol information where possible. The exact contents of a ‘response’ record are determined not just by the record type but also by the URI scheme of the record’s target-URI. The response record holds the content of the target-URI. The target-URI can be any Internet media type [49] like audio, message, image, model, multipart, text, or video. Please see Figure 4.2 for an example.

**Request**

A ‘request’ record [50] holds the details of a complete scheme-specific request, including network protocol information where possible. The exact contents of a ‘request’ record are
Figure 4.2: Example of a ‘response’ record in a WARC file

WARC/1.0
WARC-Type: response
WARC-Target-URI: http://www.youtube.com/watch?v=H4Mx5qbgeNo
WARC-Date: 2013-04-15T21:30:52Z
WARC-Payload-Digest: sha1:5DVEJSBN3F5DMNGCEIHM6HUP5XY3EO
WARC-IP-Address: 74.125.224.73
WARC-Record-ID: <urn:uuid:af4800e3-f576-4412-a5dd-daf23b02ec83>
Content-Type: application/http; msgtype=response
Content-Length: 196912

HTTP/1.0 200 OK
Date: Mon, 15 Apr 2013 21:30:52 GMT
Server: gwiseguy/2.0
X-Frame-Options: SAMEORIGIN
X-YouTube-Other-Cookies: VISITOR_INFO1_LIVE=K4RUrGXZa9s;PREF=
P3P: CP="This is not a P3P policy! See http://support.google.com/a
X-Content-Type-Options: nosniff
Cache-Control: no-cache
Expires: Tue, 27 Apr 1971 19:44:06 EST
X-YouTube-Experiment: ;931307;901439;919358;916613;932000;932004;9
Content-Type: text/html; charset=utf-8
X-XSS-Protection: 1; mode=block

<!DOCTYPE html><html .....
Figure 4.3: Example of a ‘request’ record in a WARC file

WARC/1.0
WARC-Type: request
WARC-Target-URI: http://www.youtube.com/watch?v=H4Mx5qbgeNo
WARC-Date: 2013-04-15T21:30:52Z
WARC-Concurrent-To: <urn:uuid:af4800e3-f576-4412-a5dd-daf23b02ec83
WARC-Record-ID: <urn:uuid:51497051-2eba-4c4c-9700-7c0a86e5cb7b>
Content-Type: application/http; msgtype=request
Content-Length: 590

GET /watch?v=H4Mx5qbgeNo HTTP/1.0
User-Agent: Mozilla/5.0 (compatible; special_archiver; Archive-It;
+http://archive-it.org/files/site-owners-special.html)
Connection: close
Referer: http://www.reddit.com/r/news/comments/1cen3t/
 there_was_just_an_explosion_at_the_boston/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Host: www.youtube.com

determined not just by the record type but also by the URI scheme of the record’s target-URI [50]. Please see Figure 4.3 for a example.

Metadata

A ‘metadata’ record contains content created in order to further describe, explain, or accompany a harvested resource, in ways not covered by other record types. A ‘metadata’ record will almost always refer to another record of another type, with that other record holding original harvested or transformed content. (However, it is allowable for a ‘metadata’ record to refer to any record type, including other ‘metadata’ records.) Any number of metadata records MAY reference one specific other record. The records are associated with inlinks and
Figure 4.4: Example of a ‘metadata’ record in a WARC file

```
WARC/1.0
WARC-Type: metadata
WARC-Target-URI: http://www.youtube.com/watch?v=H4Mx5qbgeNo
WARC-Date: 2013-04-15T21:30:52Z
WARC-Concurrent-To: <urn:uuid:af4800e3-f576-4412-a5dd-daf23b02ec83
WARC-Record-ID: <urn:uuid:74327770-8b5f-4a0f-b5a5-00a41aae2660>
Content-Type: application/warc-fields
Content-Length: 12663

seed:
fetchTimeMs: 261
charsetForLinkExtraction: UTF-8
outlink: http://www.youtube.com/favicon.ico I =INFERRED_MISC
outlink: http://s.ytimg.com/yts/cssbin/www-core-vfl_dHXKK.css E li
outlink: http://r1---sn-huojp-5cwe.c.youtube.com/crossdomain.xml X =JS_MISC
outlink: http://www.youtube.com/opensearch?locale=en_US E link/@href
outlink: http://s.ytimg.com/yts/img/favicon-vfldLzJxy.ico E link/@href
outlink: http://s.ytimg.com/yts/img/favicon_32-vflWoMFGx.png E link/@href
outlink: http://www.youtube.com/watch?v=H4Mx5qbgeNo E link/@href
outlink: http://m.youtube.com/watch?v=H4Mx5qbgeNo E link/@href
outlink: http://youtu.be/H4Mx5qbgeNo E link/@href
....
```

outlinks of the HTML file [51]. Please see Figure 4.4 for an example.

### 4.3 ETL and sub-Workflows

The Extract, Transform, Load (ETL) workflow [52] allows ingesting content and running machine learning tasks while indexing data at every stage in the workflow.

A full ETL workflow in LWBD as shown in Figure 4.6 consists of several sub-workflows. All
Figure 4.5: Example of a curl command executing a full ETL workflow for processing WARC files

curl -u administrator:foo -X POST -H 'Content-type:application/json' -d \
'{"inputDir":"hdfs://LucidN3:50001/input/BostonBombing/*.warc.gz", \
"inputType":"application/html", "collection":"BostonBombing", \
"mapperClass":"com.lucid.sda.hadoop.ingest.WarcIngestMapper", \
"doKMeans":"true", "doAnnotations":"true", "doSIPs":"true", "doSimdoc":"true"}' \
http://LucidN3:8341/sda/v1/client/workflows/_etl

these sub-workflows can be run together at once using the ETL workflow. Alternatively, each sub-workflow can be run independently, but some of the sub-workflows are dependent on other sub-workflows. The workflows are XML files located on HDFS that are pre-defined and used by Oozie to run Hadoop jobs.

The Ingest (see Figure 4.7) and Extract (see Figure 4.8) sub-workflows are run by default, whereas, to run any other sub-workflows, the appropriate boolean parameter needs to be ‘true’. All the workflow jobs are run as CURL commands with parameters.

The ‘collection’ parameter needs to be provided for every workflow or sub-workflow. This parameter is used to distinguish data between different collections. It also is used by Solr to index data into a specific collection. Before running any ETL workflow or any particular sub-workflow, the collection needs to be created if it does not exist already.

For every sub-workflow we show an example of the data that is generated for every sub-workflow. We consider one of the webpages we have in our collection ‘http://www.foxnews.com/us/2013/04/16/explosion-reported-near-finish-line-boston-marathon-spokesman-says/’ and show how the record content/metadata is generated in each sub-workflow.
The sub-workflows present inside an ETL workflow are:

4.3.1 Ingest sub-workflow

We need to load the data into HBase to perform any kind of operation. The Ingest workflow [53] (see Figure 4.6) takes documents and stores them in HBase while synchronizing the documents with Solr for search. In CTRnet, we use WARC files as input to the Ingest workflow and the documents retrieved from WARC files are loaded into HBase.

The software uses the WARC Hadoop library provided by the LEMUR project [54] for MapReduce tasks. The WARC files are initially loaded into the HDFS and provided as input by using ‘warcIngestMapper’ as mapperClass.

During the map-reduce task, every record of type ‘response’ in a WARC file is considered
Figure 4.7: Ingest workflow overview

Dataflow of a single map task for a single WARC record
Table 4.2: Example: Data generated during ingest sub-workflow

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>warc_date</td>
<td>2013-04-18T01:12:21Z</td>
</tr>
<tr>
<td>warc_WARC-IP-Address</td>
<td>23.61.194.209</td>
</tr>
<tr>
<td>warc_hostname</td>
<td>foxnews.com</td>
</tr>
<tr>
<td>collection</td>
<td>3645</td>
</tr>
<tr>
<td>warc_WARC-Payload-Digest</td>
<td>sha:TPQHNFJVC43GQIHNGRO652MBBBKVV</td>
</tr>
<tr>
<td>digest_shal</td>
<td>9be076953517366820ed8ce68bbddd30210aad5</td>
</tr>
<tr>
<td>Raw content</td>
<td>&lt;!DOCTYPE HTML&gt;............ &lt;/html&gt;</td>
</tr>
</tbody>
</table>

and if the content-type of that record is of media type HTML and the HTTP status is 200, then the raw content of the HTML and WARC metadata are added as an HBase document. The key of the HBase document is formed by joining the collection name and the unique ID of the WARC record (WARC-Record-ID).

The WARC file is integrated with many kinds of resources that are crawled from webpages. They range from images and videos to Javascript files. We consider only resources that are of type HTML and add them to HBase.

Table 4.2 shows an example of raw content and metadata ingest into HBase for the resource http://www.foxnews.com/us/2013/04/16/explosion-reported-near-finish-line-boston-marathon-spokesman-says/, that is archived into a WARC file. The ‘collection’ field generated in this workflow helps to distinguish which Archive-It collection the resource belongs to. We would be ingesting WARC files from all types of collections in Archive-It and maintain one large index.
4.3.2 Extract sub-workflow

The extract workflow [53] (see Figure 4.7) extracts text from any raw documents in HBase and parses it using Tika [55], then saves the text back to HBase. We use Tika to process the raw HTML documents that are present in HBase and to extract text and metadata from the raw documents.

The Apache Tika toolkit detects and extracts metadata and structured text content from various documents using existing parser libraries. Tika can extract all text from an HTML page and also extract all tags that are defined within a page. The metadata extracted by Tika is very helpful since it can be used to add rich metadata to a document.

Though Tika is very helpful for extracting metadata, not all websites have tags well defined and some websites even have non-relevant content within the tags. Different domains like to use different tags for representing the metadata, and as a result there are different tags that represent similar properties. We group all the tags in Solr, and have one field representing all the similar tags by using the ‘copyField’ property of Solr.

Content Handler

Content Handlers are used in Tika to strip text from the HTML page. The HTML pages include embedded code, scripts, text, and hyperlinks. In pages like news articles, we are interested in the actual article content rather than boilerplate text present in the HTML page. Tika full text extraction captures all the text within the HTML page, so there is much noise in the full text that we do not want to delete.

We use the Boilerpipe content handler with Tika, that is designed to detect and remove the surplus “clutter” around the main textual content of a webpage. Boilerpipe provides a
Figure 4.8: Extract workflow overview

Dataflow of single Map task for a single document

1. Get raw content of HTML
2. Extract HTML text and metadata using Tika with Boilerpipe
3. Update HBase Document
4. Output
library for news article extraction, and much more. Boilerplate is detected using shallow
text features, which is theoretically grounded by a stochastic text generation process from
Quantitative Linguistics. Boilerplate detection strategies are well analyzed on news, blogs,
and cross-domain sites. [56]

During the map-reduce task in the extract sub-workflow, we take the documents that are
ingested into HBase and parse the raw content using Tika with Boilerpipe and update the
HBase document with the text and metadata. Figure 4.8 summarizes the extract workflow.

Table 4.3: Example: Data generated during Tika sub-workflow

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tika_dc.publisher</td>
<td>Fox News</td>
</tr>
<tr>
<td>tika_dc.source</td>
<td>FoxNews.com</td>
</tr>
<tr>
<td>tika_keywords</td>
<td>Concord, New York City, Deval Patrick, Manhattan, Boston-Marathon, Gov Deval Patrick, Boston, NewYork City Police Department</td>
</tr>
<tr>
<td>tika_dc.description</td>
<td>At least two people are dead and 73 injured including up to 10 with amputated limbs - after two bomb tore through the finishline of the Boston Marathon, according to the Boston Police Department.</td>
</tr>
<tr>
<td>tika_description</td>
<td>At least two people are dead and 73 injured including up to 10 with amputated limbs - after two bomb tore through the finish line of the Boston Marathon, according to the Boston Police Department.</td>
</tr>
<tr>
<td>tika_og:description</td>
<td>The Boston Marathon headquarters has been locked down after an explosion was reported near the finishline, a spokesman told Reuters.</td>
</tr>
<tr>
<td>tika_prism.channel</td>
<td>fnc</td>
</tr>
<tr>
<td>tika_content-encoding</td>
<td>UTF-8</td>
</tr>
<tr>
<td>tika_content-language</td>
<td>en</td>
</tr>
<tr>
<td>tika_dc.language</td>
<td>en-US</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tika_dc.type</td>
<td>Text.Article</td>
</tr>
<tr>
<td>tika_dcterms.created</td>
<td>2013-04-1600:00:45EDT</td>
</tr>
<tr>
<td>tika_twitter:site</td>
<td>@foxnews</td>
</tr>
<tr>
<td>tika_title</td>
<td>FBI, Boston police say range of suspects, motives remains ‘wideopen’</td>
</tr>
<tr>
<td>tika_dc.title</td>
<td>FBI, Boston police say range of suspects, motives remains ‘wideopen’</td>
</tr>
<tr>
<td>tika_og:title</td>
<td>FBI, Boston police say range of suspects, motives remains ‘wideopen’</td>
</tr>
<tr>
<td>tika_content-type</td>
<td>text/html;charset=UTF-8</td>
</tr>
<tr>
<td>tika_dc.date</td>
<td>2013-04-16T00:00:00Z</td>
</tr>
<tr>
<td>tika_dc:title</td>
<td>FBI, Boston police say range of suspects, motives remains ‘wideopen’</td>
</tr>
<tr>
<td>tika_og:type</td>
<td>article</td>
</tr>
<tr>
<td>tika_dcterms.modified</td>
<td>2013-04-1600:00:45EDT</td>
</tr>
<tr>
<td>tika_dcterms.abstract</td>
<td>At least two people are dead and 73 injured including up to 10 with amputated limbs - after two bomb tore through the finish line of the Boston Marathon, according to the Boston Police Department.</td>
</tr>
<tr>
<td>tika_dc.subject</td>
<td>Concord, New York City, Deval Patrick, Manhattan, Boston Marathon, Gov Deval Patrick, Boston, New York City Police Department</td>
</tr>
<tr>
<td>tika_og:site_name</td>
<td>Fox News</td>
</tr>
<tr>
<td>tika_twitter:card</td>
<td>summary</td>
</tr>
<tr>
<td>tika_prism.aggregationtype</td>
<td>subsection</td>
</tr>
</tbody>
</table>
Table 4.3 – continued from previous page

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tika_fb:app_id</td>
<td>113186182048399</td>
</tr>
<tr>
<td>tika_dc.identifier</td>
<td>urn:uuid:d3eb762250f0e310VgnVCM100000d7c1a8c0RCRD</td>
</tr>
<tr>
<td>Text</td>
<td>FBI, Boston police say range of suspects, motives remains ‘wide open’</td>
</tr>
</tbody>
</table>

Table 4.3 shows the Tika metadata and text that are generated for the news article. The LWBD software adds a prefix ‘tika_’ for all the fields generated by Tika software. ‘http://www.foxnews.com/us/2013/04/16/explosion-reported-near-finish-line-boston-marathon-spokesman-says/’.

Digital news publishers use different types of date formats within the meta-tags in the webpage. Solr uses ISO date format as input for date field types. We converted all the dates (see Table 4.4 for examples) extracted through Tika to the ISO format using the natty library [57]. After running the tagcount on the metadata produced by Tika for all the HTML documents, we found 19 metadata tags (See Figure 4.9) that are commonly used to represent the dates.
Table 4.4: Example: Date formats before and after conversion to ISO 8601 format

<table>
<thead>
<tr>
<th>Before Conversion</th>
<th>After Conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012/11/09</td>
<td>2012-11-09T00:00:00Z</td>
</tr>
<tr>
<td>2012-11-10</td>
<td>2012-11-10T00:00:00Z</td>
</tr>
<tr>
<td>2012-11-10T17:15</td>
<td>2012-11-11T01:15:00Z</td>
</tr>
<tr>
<td>November 25, 2012, 12:42 pm</td>
<td>2012-11-25T20:42:00Z</td>
</tr>
<tr>
<td>Sunday, 04 September 2011 19:09:53 PDT</td>
<td>2011-09-05T02:09:53Z</td>
</tr>
<tr>
<td>Tue, 23 Aug 2011 16:08:45 EST</td>
<td>2011-08-23T20:08:45Z</td>
</tr>
</tbody>
</table>

Figure 4.9: Metadata tags used to represent dates by different Web publishers

- pubtime, publishdate, publish_date, publication_date, pubdate,
- article_date_original, article_date_updated, pubdatetime, displaydate,
- dc_date_issued, dc_date, created, dcterms_created, og:created_time,
- originalpublicationdate, pd, dcterms_modified, last-modified

4.3.3 Vectorize sub-workflow

The vectorize sub-workflow [58] (see Figure 4.10) prepares ingested and extracted content for machine learning tasks like document similarity or statistically interesting phrases. The documents are transformed into TF-IDF vectors suitable for machine learning. This workflow is run before any of the collocations, annotations, k-means, or similar document workflows can be run.

The vectors are saved into a sequence file on HDFS, which is later used for other sub-workflows.

4.3.4 Annotations sub-workflow

The annotations sub-workflow [59] (see Figure 4.11) extracts entities from vectorized content and adds them as fields to the documents. The annotations implementation uses a PEAR
Figure 4.10: Vectorize workflow overview

- **Input**: HBase
- **Documents**: Collection
  - documentsAsText
  - documentsAsVectors
  - ven_nGrams
  - vec_analyzer
  - zkConnect
  - parentWfId

- **Data splits**: Map tasks
  (Dump data to HDFS)
- **Output**:
  - Mahout Map Reduce job
    for generating Sparse vectors
  - TF-IDF vectors

- **HDFS**: Sequence file
- **Vectorize Sub-wf**: Collection
  - documentsAsText
  - documentsAsVectors
  - ven_nGrams
  - vec_analyzer
  - zkConnect
  - parentWfId

- **Sequence file**

---

**Sequence file**:

- ...
package to run UIMA across the documents. We use the openNlp PEAR package to extract annotations from the text.

For every document, the annotations ‘person’, ‘location’, and ‘organization’ are extracted.

4.3.5 K-Means sub-workflow

The K-Means sub-workflow [60] (see Figure 4.12) creates k-means clusters. Clustering allows identifying similar documents using a K-Means algorithm to determine documents that can be grouped together.

K-Means clustering

K-Means is a rather simple but well known algorithm for grouping objects, i.e., clustering. Again all objects need to be represented as a set of numerical features. In addition the user has to specify the number of groups (referred to as k) he wishes to identify.

Each object can be thought of as being represented by some feature vector in an n dimensional space, n being the number of all features used to describe the objects to cluster. The algorithm then randomly chooses k points in that vector space; these points serve as the initial centers of the clusters. Afterwards all objects are each assigned to the center they are closest to. Usually the distance measure is chosen by the user and determined by the learning task.

After that, for each cluster a new center is computed by averaging the feature vectors of all objects assigned to it. The process of assigning objects and recomputing centers is repeated until the process converges. The algorithm can be proven to converge after a finite number of iterations. [61]
Figure 4.11: Annotate workflow overview

Dataflow in a Single map task for a single document
The default values of kmeans_convergenceDelta, kmeans_maxIter, and kmeans_numClusters are 0.5, 10, and 20 respectively. We should configure these parameters based on the data set and can keep re-configuring the parameters until we find the right configuration for forming good clusters.

As shown in Table 4.5, the map-reduce task in K-Means adds the fields ‘clusterId’ and ‘distanceToCentroid’ to each document in HBase.
Table 4.5: Example: Data generated during K-Means sub-workflow

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>distanceToCentroid</td>
<td>0.44192549146</td>
</tr>
<tr>
<td>clusterId</td>
<td>21246</td>
</tr>
</tbody>
</table>

4.3.6 Similar Documents sub-workflow

The “simdoc” workflow identifies documents that are similar to each other. The result is a set of scores between 0 and 1 for the level of similarity to other documents.

As shown in Figure 4.13, the simdoc sub-workflow runs Matrix conversion and RowSimilarity jobs in Apache Mahout. Once the scores are calculated, they are updated into HBase. The updated values in HBase are not indexed into Solr and are only accessible through HBase. Cosine similarity is used to calculate the scores.

Table 4.6 shows an example of Similar Documents sub-workflow output for the document http://foxnews.com/us/2013/04/16/explosion-reported-near-finish-line-boston-marathon-spokesman-says/. A score of 1.0 points out the exact duplicates; scores are in descending order of similarity.

4.3.7 Collocations sub-workflow

The collocations sub-workflow [62] performs analysis to produce a list of statistically interesting phrases (SIPs). The SIPs are not saved into HBase like any other sub-workflow but they are indexed directly into Solr.

The collocations sub-workflow has two parts:

1. Generate the collocations (SIPs workflow)
Figure 4.13: Similar Documents workflow overview

Input

TF-IDF Vectors

HBase

Simdoc Sub-wf

numberOfRegions

collectionID

parentWfId

Simdoc_threshold

zkConnect

Data splits

Map-Reduce jobs
(Mahout RowSimilarity Job)

Output

HDFS

Map-Reduce jobs
(Sequence file to Map file)

Output

Map Jobs
(Send documents to Hbase)

Output
Table 4.6: Example: Data generated during Sim Doc sub-workflow

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="">urn:uuid:a19c3e45-f67c-4d84-afae-bdc864af6810</a></td>
<td>1.0</td>
</tr>
<tr>
<td><a href="">urn:uuid:01213c44-be4a-4bf3-6d52-6df701641340</a></td>
<td>1.0</td>
</tr>
<tr>
<td><a href="">urn:uuid:acae8e8cb-782-452-8a24-9e25276b601c</a></td>
<td>0.777425412341</td>
</tr>
<tr>
<td><a href="">urn:uuid:260feaa3-5f90-4f13-9739-617c344d8433</a></td>
<td>0.766599702408</td>
</tr>
<tr>
<td><a href="">urn:uuid:2b4b2ae4-d5e9-4309-960a-9cc14a7f7da</a></td>
<td>0.765531168833</td>
</tr>
<tr>
<td><a href="">urn:uuid:1cba4fe-36ca-4017-86a2-2e7fd63389f1</a></td>
<td>0.763054024898</td>
</tr>
<tr>
<td><a href="">urn:uuid:f96b19ae-b3da-41f8-9ec3-29aebe6613a8</a></td>
<td>0.741529643087</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td><a href="">urn:uuid:54f297d4-c523-484a-9fc8-bef15f9e4fe3</a></td>
<td>0.593956610373</td>
</tr>
<tr>
<td><a href="">urn:uuid:451da40-6679-4b67-9a8a-34f66d102a3e</a></td>
<td>0.593956610373</td>
</tr>
<tr>
<td><a href="">urn:uuid:21cd4ff5-cc5-4707-9650-776553c5eb36</a></td>
<td>0.593774841814</td>
</tr>
</tbody>
</table>

2. Index the collocations (Index_sips workflow)

**SIPs workflow**

The SIPs workflow (see Figure 4.14) takes input as vectorized documents from HDFS and the map-reduce task generates the collocations and saves them back onto HDFS.

**Index-SIPs workflow**

The Index-SIPS workflow (see Figure 4.15) takes the generated collocations from HDFS and then indexes the collocations to Solr or SolrCloud. If the ‘solr_zkhost’ parameter is set, then the data is indexed to SolrCloud, otherwise data is indexed to a single Solr instance.
Figure 4.14: SIPS workflow overview

- **Input**: Vectorized Documents
- **Sips Sub-wf**: parentWfId, Input, Output, Sips_minLLR, Sips_maxNGramSize
- **Data splits**
- **Map-Reduce jobs**: (Mahout Colloc Driver)
- **Output**: HDFS
Figure 4.15: Index-SIPS workflow overview

- Input: HDFS
- Output: Solr
- Data splits
- Map-Reduce jobs (Index to Solr or SolrCloud)
- Solr

Fields:
- solrUrl
- solrFailedThresholdPercent
- solrZKHost
- solrZkCollection
Chapter 5

CTRnet Digital Library

5.1 5S Perspective

5S (Streams, Structures, Spaces, Scenarios, and Societies) is a unified formal theory for Digital Libraries (DLs). Figure 5.1 begins with the 5Ss, and key concepts of a DL (e.g., digital object, collection), and then adds in about CTRnet digital objects and other concepts we built into the CTRnet-DLA (Digital Library Archive). The arrows in the figure represent dependencies, indicating that a concept is formally defined in terms of previously defined concepts that point to it. In the CTRnet DLA, we now look at how each of the concepts maps to the 5Ss.

5.1.1 Societies

The CTRnet Digital Library includes different types of societies from archiving and accessing points of view. For example, Archivists, Librarians, and Researchers will be interested in collecting event specific information and helping to create permanent archives. Including
Figure 5.1: 5S definitional structure extended for CTRnet. Adapted from [3]
the groups mentioned, the public sector, Government agencies, and social service providers (e.g., Non-profit organizations) will be interested in accessing the archived and processed information.

5.1.2 Streams

Any kind of Internet media is considered as a stream in this Digital Library, in part since the content is created by crawling resources from the Web. Though the societies can be interested in a specific stream (like text or image or video), all types of media are collected and archived. Further, workflows and sub-workflows map streams of data to streams of results. Streams flow into and from the storage systems and nodes. Result streams also are presented to users.

5.1.3 Scenarios

From the scenarios point of view, user scenarios describe one or more users engaged in some meaningful activity with the CTRnet system like fact-finding, learning, gathering and exploring through the structures and services (e.g., Annotation, Searching, Clustering, Browsing). Figure 5.2 shows the taxonomy of CTRnet Digital Library services.

5.1.4 Structures

Though the WARC files that are used as input are structured records, the documents present within the files can be semi-structured (e.g., HTML). We try to extract structure out of the semi-structured documents by processing the documents using LucidWorks Big Data software and by creating additional metadata for the documents.
Figure 5.2: Taxonomy of CTRnet Digital Library Services. Adapted from [4].

<table>
<thead>
<tr>
<th>Infrastructure Services</th>
<th>Information Satisfaction Services</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Repository Building</strong></td>
<td><strong>Add Value</strong></td>
</tr>
<tr>
<td><strong>Creational</strong></td>
<td><strong>Preservational</strong></td>
</tr>
<tr>
<td>Acquiring</td>
<td>Converting</td>
</tr>
<tr>
<td>Crawling</td>
<td>Replicating</td>
</tr>
<tr>
<td>Harvesting</td>
<td>Clustering</td>
</tr>
<tr>
<td></td>
<td>Extracting</td>
</tr>
<tr>
<td></td>
<td>Indexing</td>
</tr>
<tr>
<td></td>
<td>Linking</td>
</tr>
<tr>
<td></td>
<td>Browsing</td>
</tr>
<tr>
<td></td>
<td>Filtering</td>
</tr>
<tr>
<td></td>
<td>Providing access</td>
</tr>
<tr>
<td></td>
<td>Searching</td>
</tr>
</tbody>
</table>

### 5.1.5 Spaces

In the CTRnet Digital Library, we distinguish spaces by the operations on the documents. The set of spaces that we use in this digital library are involved in Indexing, Annotating, Indexing, Clustering, Browsing, Searching, and Extraction. The vectorizing sub-workflow produces the key spaces used. The user interface makes use of 2D spaces for presentation.

### 5.2 Case Study: CTRnet DLA Prototype

We considered the Boston Marathon Bombings collection for building a Digital Library prototype using Archive-It and LucidWorks Big Data software. The seeds are collected through tweets and through the Google Document created by the Internet Archive. We downloaded WARC files from the first 4 days of the collected data and processed them using the LucidWorks software set up on the cluster.

After all the data is loaded into HDFS, a full ETL workflow was executed. All the processed data is indexed into a Solr collection, which is retrieved for the user interface through the
Solr API. The prototype is built using the Python mini framework CherryPy [41] and using the template library Jinja2 [42].

Figure 5.3 shows the home page for the Boston Marathon Bombings collection.

5.2.1 Facets

The metadata that is extracted from the documents (e.g., article date, hostname, keywords, etc.) are used as facets that help users filter through the data set based on their selection.
Table 5.1: WARC statistics for ‘Boston Marathon Bombings’ collection

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of WARC files</td>
<td>50</td>
</tr>
<tr>
<td>Size of collection</td>
<td>47 GB</td>
</tr>
<tr>
<td>Number of documents</td>
<td>23596</td>
</tr>
</tbody>
</table>

Figure 5.4 shows the placement of facets and HTML documents.

5.2.2 Clustered Documents

The text in the HTML documents that is extracted using Boilerpipe is used as input for the K-Means clustering component in the LucidWorks software. Instead of showing all the search results, we presented the user only a document from each cluster and let the user explore more articles in the cluster by using the ‘Find Related Articles’ link. This is implemented using the Solr Grouping feature.

Figure 5.5 shows a screenshot of the cluster groups in the prototype.

5.2.3 LWBD Sub-workflow timings

Table 5.2 shows the time taken by each sub-workflow in LWBD for the ‘Boston Marathon Bombings’ collection. Out of all the sub-workflows, Mahout jobs (Annotation, Similar Documents) are time consuming jobs. The machines used within the cluster only have a total of 7GB memory available each and more than 75% of this available memory is consumed by the tasktracker, sda-events, regionserver, datanode. This resulted in less memory available for the running Mahout jobs and the jobs took longer time to finish.
Figure 5.4: Facets and HTML documents in the Digital Library
Figure 5.5: Using cluster groups in the Digital Library
Table 5.2: LWBD Sub-workflow timings for ‘Boston Marathon Bombings’ collection

<table>
<thead>
<tr>
<th>Sub-workflow name</th>
<th>Time (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingest</td>
<td>50</td>
</tr>
<tr>
<td>Extract</td>
<td>13</td>
</tr>
<tr>
<td>Vectorize</td>
<td>18</td>
</tr>
<tr>
<td>K-Means</td>
<td>80</td>
</tr>
<tr>
<td>SIPs</td>
<td>07</td>
</tr>
<tr>
<td>IndexSIPs</td>
<td>49</td>
</tr>
<tr>
<td>Annotate (6 job tasks failed at 91%, only 21K documents processed)</td>
<td>90</td>
</tr>
<tr>
<td>Similar Documents</td>
<td>695</td>
</tr>
</tbody>
</table>

5.2.4 Lessons Learned

1. LWBD provided a very good framework for processing and indexing WARC files using Hadoop and Solr. Using the framework, we could do real-time processing of WARC files over large volumes and serve the documents to users using Solr index.

2. Over the course of this thesis, we are limited by the availability of good hardware to set up a powerful cluster. This resulted in slow Hadoop jobs which affected the processing of large datasets. Instead of using 30 machines for the cluster with 8GB, it will be very effective to use 15 machines with more memory.

3. Maintenance of the cluster over 30 machines consumed much time. The non-availability of single GUI type interface for monitoring cluster status is a limitation. In the future, tools like Zabbix [63] should be used to monitor cluster status.

4. Bugs present in LWBD software in the 1.1 version limited us in taking advantage of features like Annotation. These bugs were reported and resolved in the later versions.
Chapter 6

Conclusions and Future work

6.1 Conclusions

In this thesis, we discussed how to create archives, process the text documents within the archived files, produce metadata, and build digital library services using the indexed data and metadata. We showed how the 5S framework helps with modeling and describing the digital library, and how LWBD software can be utilized to provide DL services. The next sections discuss how the work in this thesis can be used for different stakeholders in the system.

User of my current work

To continue with the current work, a user should load the WARC files incrementally into the HDFS and run the full ETL workflow. The user should use the same collection name, so that by the end all the data is covered by a single index and the stakeholders will be able to search across collections. Once the data is indexed, a new user interface can be built.
using the REST API provided by LucidWorks software or the current scripts (CherryPy and Jinja2) can be used to present data in the user interface.

**Maintenance of my current work**

The local admin of the software should apply patches, to customize the LucidWorks software for processing WARC files. The software should be properly tested to make sure all the components are functioning as expected and the provision software should be used to monitor the nodes in the cluster and the disk space/load.

**LucidWorks Customer**

The digital library described in this thesis can be used by any LucidWorks customer interested in processing WARC/HTML files.

### 6.2 Future Work

In this section, we discuss future work that can be built upon this thesis.

**CTRnet/IDEAL follow on**

The CTRnet project and the follow on IDEAL project (i.e., the sequel to CTRnet) might involve carrying out some or all of the following:

- Process the 10 TB of WARC files that represent the full set of data collected by CTRnet and provide access using the LucidWorks Big Data software.
• Image and video metadata also should be extracted from the HTML/WARC files using Tika annotation.

• Crowd sourcing or focused groups can be used for selecting labels for the clusters.

• The tweet dump from MySQL should be processed using LucidWorks software and the data should be indexed which will be helpful to generate faster visualizations for tweets.
Bibliography


library for rescue and recovery (winner of best poster in category award).” International Outreach NOW Conference, September 2009.


