International Journal of Scientific & Engineering Research, Volume 6, Issue 9, September-2015
ISSN 2229-5518

Comparison of Open Source Crawlers- A Review

Monika Yadav, Neha Goyal

Abstract— Various open source crawlers can be characterized by the features they implement as well as the performance they have in different scenario. This paper will include the comparative study of various open source crawlers. Data collected from different websites, conversations and research papers shows the practical applicability of open source crawlers.

Index Terms— Comparison, Key features, Open source crawlers, Parallelization, Performance, Scale and Quality.

1 INTRODUCTION

THIS research paper aims at comparison of various available open source crawlers. Various open source crawlers are available which are intended to search the web. Comparison between various open source crawlers like Scrapy, Apache Nutch, Heritrix, WebSphinix, JSpider, GnuWget, WIRE, Pavuk, Teleport, WebCopier Pro, Web2disk, WebHTTrack etc. will help the users to select the appropriate crawler according to their needs.

This study will includes the discussion of various quality terms for different open source crawlers. A brief of various quality terms like Freshness, Age, Communication Overhead, coverage, Quality, Overlap is taken into consideration. Various techniques of crawling and their effect on these quality terms have been discussed [3]. Then comparison of different open source crawlers is done in terms of key features, Language, Operating system, License, Parallel. An experiment shows the comparison of different crawlers in terms of related words, depth and time [8]. Various statistics collected by V.M. Preito et al. [9] about types of links visited and proposed scale of various open source crawlers.

Different users or organizations uses the crawlers for different purpose some require the fast result, some concentrate on scalability, some needed the quality and others requires less communication overhead.

One can compromise other depending upon their requirement. Comparison will help to decide which crawler is suitable to them.

2 GENERAL DESCRIPTION OF OPEN SOURCE CRAWLERS

2.1 Properties of open source crawlers

Various properties that a web crawler must satisfy are:

- Robustness: Crawlers must be designed to be resilient to trap generated by various web servers which mislead the crawlers into getting stuck fetching infinite number of pages in particular domain. Some such traps are malicious which results in faulty website development.
- Politeness: web servers have some set of policies for crawlers which visit them in order to avoid overloading websites.
- Distributed: The crawler should have the ability to execute in a distributed fashion across multiple machines.
- Scalable: The crawler architecture should permit scaling up the crawl rate by adding extra machines and bandwidth.
- Performance and efficiency: The crawl system should make efficient use of various system resources including processor, storage and network band- width.
- Quality: Quality defines how important the pages are, downloaded by crawlers. Crawler tries to download the important pages first.

- *Monika Yadav has done masters degree program in Computer Science Eng. From Banasthali University, Rajasthan, India, PH-01438228477. E-mail: mnk.yadav722@gmail.com.*
- *Neha Goyal is currently pursuing PHD in Computer Science Eng. from The Northcap University, India, PH-01242365811. E-mail: nehagoyal@ncuindia.edu*

International Journal of Scientific & Engineering Research, Volume 6, Issue 9, September-2015
ISSN 2229-5518

- Freshness: In many applications, the crawler should operate in continuous mode: it should obtain fresh copies of previously fetched pages. A search engine crawler, for instance, can thus ensure that the search engine's index contains a fairly current representation of each indexed web page. For such continuous crawling, a crawler should be able to crawl a page with a frequency that approximates the rate of change of that page.
- Extensible: Crawlers should be designed to be extensible in many ways – to cope with new data formats, new fetch protocols, and so on. This demands that the crawler architecture be modular.

2.2 Techniques of crawling and their effect on various parameters

Various techniques have been used in searching the web by web crawler. Some of the searching techniques, their objective and factors are mentioned in Table. During the crawl, there is cost associated with not detecting the event and thus having an outdated copy of resource. The most used cost functions are Freshness and age.

Freshness: it indicate that whether the local copy is accurate or not. The freshness of page p in the repository at time t is defined as:

$$F_p(t) = \begin{cases} 1 & \text{If p is equal to the local copy at time t} \\ 0 & \text{Otherwise} \end{cases}$$

Age: It indicates how outdated the local copy is. The age of a page p in the repository at time t is defined as:

$$A_p(t) = \begin{cases} 1 & \text{if p is not modified at time t} \\ t - & \text{modification time of p} \\ & \text{otherwise} \end{cases}$$

Age provides the data for dynamicity as shown in table.

Apart from these cost functions there are various quality terms like coverage, Quality and communication overhead which are considered during performance of parallelization.

Various parameters for measuring performance of parallelization are:

Communication Overhead:

In order to coordinate work between different partition, parallel crawler exchange messages. To quantify how much communication is required for this exchange, communication overhead can be defined as the average number of inter partition URLs exchanged per downloaded page.

Communication overhead can be defined as:

$$\frac{U}{N}$$

Where

U= No. of exchanged inter-partition URLs by parallel crawler.

N= Total no. of downloaded pages by parallel crawler.

Overlap:

Overlap may occur when multiple parallel crawler download the same page multiple times.

Overlap can be defined as:

$$\frac{N - I}{I}$$

Where N represents the total number of pages downloaded by the overall crawler, and I represents the number of unique pages downloaded.

Quality:

Quality defines how important the pages are, downloaded by crawlers. Crawler tries to download the important pages first.

Quality can be defined as:

$$\frac{|A_N \cap P_N|}{|P_N|}$$

If a crawler download N important pages in total, PN to represent that set of N pages. We also use AN to represent the set of N pages that an actual crawler would download, which would not be necessarily the same as PN. Importance and relevance can be defined in terms of quality as mentioned in Table.

Coverage:

It is possible that all pages are not downloaded by parallel crawlers that they have to, due to lack of inter communication.

Coverage can be defined as:

International Journal of Scientific & Engineering Research, Volume 6, Issue 9, September-2015
ISSN 2229-5518

$$\frac{I}{U}$$

Where U represents the total number of pages that the overall crawler has to download, and I is the number of unique pages downloaded by the overall crawler. Three crawling modes are also used to measure these parameters of performance of parallelization which are described below:

Firewall mode: In this mode, pages are downloaded only within its partition by parallel crawler and it does not follow any inter-partition link. All inter-partition links are ignored and thrown away.

In this mode, the overall crawler does not have any overlap in the downloaded pages, because a page can be downloaded by only one parallel crawler. However, the overall crawler may not download all pages that it has to download, because some pages may be reachable only through inter-partition links.

Cross Over mode: A parallel crawler downloads pages within its partition, but when it runs out of pages in its partition, it also follows inter-partition links

In this mode, downloaded pages may clearly overlap, but the overall crawler can download more pages than the firewall mode. Also, as in the firewall mode, parallel crawlers do not need to communicate with each other, because they follow only the links discovered by them.

Exchange mode: When parallel crawlers periodically and incrementally exchange inter-partition URLs, we say that they operate in an exchange mode. Processes do not follow inter-partition links. In this way, the overall crawler can avoid overlap, while maximizing coverage.

Table 1. shows the role of these three modes in performance of parallelization. Where "good" means that the mode is expected to perform relatively well for that metric and "Bad" means that it may perform worse compared to other modes.

TABLE I
COMPARISON OF THREE CRAWLING MODES

MODE	COVERAGE	OVERLAP	QUALITY	COMMUNICATION
FIREWALL	BAD	GOOD	BAD	GOOD
CROSS-OVER	GOOD	BAD	BAD	GOOD
EXCHANGE	GOOD	GOOD	GOOD	BAD

Table 2. shows the various techniques of searching used by crawlers in terms of coverage, freshness, importance, relevance and dynamicity.

TABLE 2
TAXONOMY OF CRAWL ORDERING TECHNIQUE [3]

Techniques	Objectives		Factors Considered		
	Coverage	Freshness	Importance	Relevance	Dynamacity
Breath First Search	Yes	-	-	-	-
Prioritize by indegree	Yes	-	Yes	-	-
Prioritize by PageRank	Yes	-	Yes	-	-
Prioritize by Site Size	Yes	-	Yes	-	-
Prioritize by Spawning rate	Yes	-	-	-	Yes
Prioritize by search impact	Yes	-	Yes	Yes	-
Scooped crawling	Yes	-	-	Yes	-
Minimize Obsolescence	-	Yes	Yes	-	Yes
Minimize age	-	Yes	Yes	-	Yes
Minimize incorrect content	-	Yes	Yes	-	Yes
Minimize embarrassment	-	Yes	Yes	Yes	Yes
Maximize search impact	-	Yes	Yes	Yes	Yes
Update capture	-	Yes	Yes	Yes	Yes
Web Foundation	Yes	Yes	-	-	Yes
OPIC	Yes	Yes	Yes	-	Yes

International Journal of Scientific & Engineering Research, Volume 6, Issue 9, September-2015
ISSN 2229-5518

3 COMPARISON OF VARIOUS OPEN SOURCE CRAWLERS

Various Open Source Crawlers differ from one another in terms of scalability, Flexibility and their performance in different scenario. Adaptation of particular crawlers by user or organization totally depends on their requirement. There are some key feature differentiation is given in Table which will help the user to select the appropriate crawler according to their requirement

Some of the most useful open source crawlers like Scrapy, Apache Nutch, Heritrix, WebSphinix, JSPider, GNUWget, WIRE, Pavuk, Teleport, WebCopier Pro, Web2disk and WebHTTrack based on data from various websites and conversations have been discussed in this research paper.

3.1 Key features of open source crawlers

Key Features of Scrapy:

1. Easy to setup and use if you know Python.
2. Faster than Mechanize but not as scalable as Nutch or Heritrix.
3. Excellent developer documentation.
4. Built-in JSON, JSON lines, XML and CSV export formats.
5. Scrapy is also an excellent choice for focused crawls.
6. Scrapy is a mature framework with full unicode, redirection handling, gzipped responses, odd encodings, integrated http cache etc.

Key Features of Apache Nutch:

1. Highly scalable and relatively feature rich crawler.
2. Features like politeness which obeys robots.txt rules.
3. Robust and scalable - Nutch can run on a cluster of up to 100 machines.
4. Quality - crawling can be biased to fetch "important" pages first.
5. Plays nice with Hadoop. Besides, integrates with other parts of the Apache ecosystem like Tika and Solr.
6. All the crawl data can be stored in a distributed key value store, like HBase(as of Nutch 2.0).
7. Lots of support because of widespread use. Check the mailing lists to see what I mean.
8. Dynamically scalable (and fault-tolerant) through Hadoop
9. Flexible plugin system
10. Stable 1.x branch

Key Features of Heritrix:

1. Excellent user documentation and easy setup
2. Mature and stable platform. It has been in production use at archive.org for over a decade
3. Good performance and decent support for distributed crawls.
4. Scalable but not dynamically scalable.
5. Ability to run multiple crawl jobs simultaneously. The only limit on the number of crawl jobs that can run concurrently is the memory allocated to Heritrix.
6. More secure user control console. HTTPS is used to access and manipulate the user control console.
7. Increased scalability. Heritrix 3.0 allows stable processing of large scale scrawls.
8. Increased flexibility when modifying a running crawl.
9. Enhanced extensibility through the spring framework. For example, domain overrides can be set at a very fine-grained level.

Key Features of WebSphinix:

1. Multithreaded Web page retrieval
2. An object model that explicitly represents pages and links
3. Supports for reusable page content classifiers
4. Support for the robot exclusion standard

Key Features of JSPider:

1. Checks sites for errors.
2. Outgoing and/or internal link checking.
3. Analyze site structure.
4. Download complete web sites

Key Features of GNUWGet:

1. Can resume aborted downloads.
2. Can use filename wild cards and recursively
3. Mirror directories.
4. Supports HTTP proxies and HTTP cookies.
5. Supports persistent HTTP connections.

Key Features of WIRE:

1. Good Scalability
2. Highly configurable i.e. all the parameters for crawling and indexing can be configured, including several scheduling policies

International Journal of Scientific & Engineering Research, Volume 6, Issue 9, September-2015
ISSN 2229-5518

3. High Performance (The downloader modules of the WIRE crawler ("harvesters") can be executed in several machines).

Key Features of Pavuk:

1. It can provide detailed timing information about transfers.
2. It can be used as a full featured FTP mirroring tool (preserves modification time, permissions, and symbolic links).
3. Optional transfer speed limitation max./min.
4. Optional multithreading support
5. Multiple round-robin used HTTP proxies.
6. Supports NTLM authorization.
7. It has JavaScript bindings to allow scripting of particular tasks.

Key Features of Teleport:

1. JavaScript parsing capability for better, more thorough exploration of complex sites.
2. Ten simultaneous retrieval threads get data at the fastest speeds possible.
3. Full offline browsing and site mirroring capabilities
4. Project Scheduler.
5. Java applet retrieval gets Java classes and base classes.
6. Retrieval filters let you download only files matching desired type and size constraints.

Key Features of WebCopierPro:

1. Projects management window
2. Multiple projects download
3. Enhanced Integrated browser
4. New DHTML and JavaScript parser
5. Project Export in multiple formats
6. Automatic project export
7. Recursive download method

Key Features of Web2disk:

1. Easy to use.
2. Fixes downloaded websites for easy offline browsing.
3. Web2disk has one time cost whether you download 20 pages or 20k pages.
4. Automatically save snapshots of your website daily, weekly or monthly.
5. Monitor websites for update. Download dynamic pages.
6. Powerful filtering.

Key Features of WebHTTrack:

1. Fully configurable, and has an integrated help system.
2. Arranges the original site's relative link-structure.
3. Easy-to-use offline browser utility.
4. It can also update an existing mirrored site, and resume interrupted downloads.

TABLE 3
COMPARISON OF OPEN SOURCE CRAWLERS IN TERMS OF VARIOUS PARAMETERS

Open source Crawlers	Language	Operating System	License	Parallel
Scrapy	Python	Linux/Mac OS X/Windows	BSD License	Yes(During broad crawls)
Apache Nutch	Java	Cross-platform	Apache License 2.0	Yes (Using Hadoop)
Heritrix	Java	Linux/Unix-like/Windows Unsupported	Apache License	Yes
WebSphinix	Java	Windows, Mac, Linux, Android, IOS	Apache Software License	Yes
JSpider	Java	Windows, Windows7, Window vista	GNU Library or Lesser General Public License version 2.0 (LGPLv2)	Yes
Gnu Wget	C	Cross-platform	GNU General Public License version 3 and later	
WIRE	C/C++		GPL LIcense	
Pavuk	C	Linux	GNU General Public License Version 2.0(GPLV2)	Yes
Teleport	-	Windows	Apache License	Yes
Web2disk	-	Windows	-	Yes
WebCopierPro	-	Windows/Mac OS X	-	No
WebHTTrack	C/C++	Cross-Platform	GPL	Yes

International Journal of Scientific & Engineering Research, Volume 6, Issue 9, September-2015
ISSN 2229-5518

3.2 Comparison in terms of Language, Operating System, License and Parallelization

Table 3. shows comparison of open source crawlers in terms of language used, Operating System, License and parallelization .

3.3 Comparison in terms of Related Words, Depth and Time

During the crawl certain factors plays key role in determining quality of a web crawlers like how much time it will take to crawl the web and brings the most important pages or no. Of related words first and performance of crawler. Fig. 1, Fig. 2 and Fig. 3 shows the related words , depth and time of various open source crawlers respectively as a result of experiment [8].

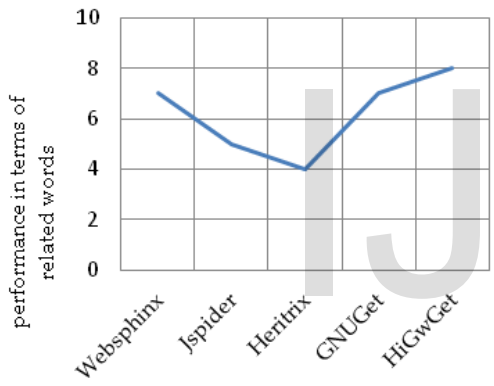

Fig.1. the Comparison of different crawlers in terms of related words [8].

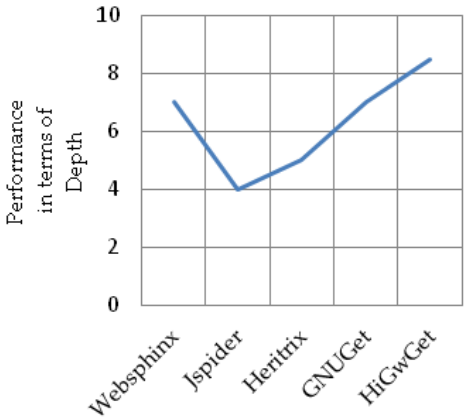

Fig. 2. The comparison of different crawlers in terms of Depth [8].

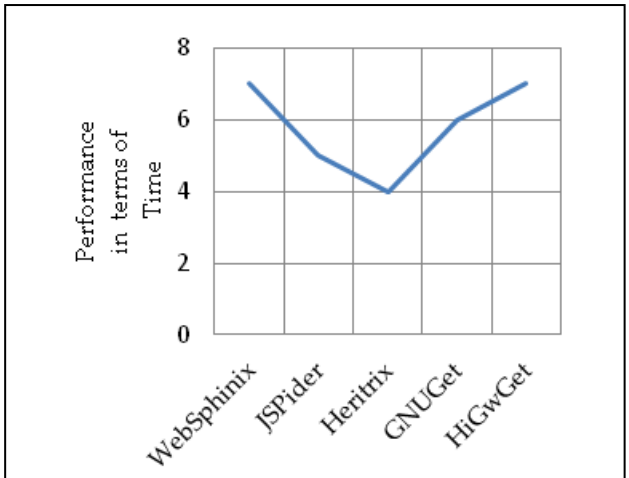

Fig. 3. The comparison of different web crawlers in terms of Time [8].

3.4 Comparison in terms of link visited and proposed scale

An experiment by V.M. Preito et al. provides the summary of various types of links processed by open source crawlers.

Various types of links like Href=Java Script Link, Document write Link, Menu Link, Flash Link, Applet Link, Redirects, Class or Java Link, Ajax Link, Links with a, VbScript Link, Static String link, Concatenated String Link, Special Function String Link, Tests Passed have been taken into consideration and then percentage of these links visited by various crawlers like Heritrix, Nutch, Pavuk, Teleport, Web2disk, WebCopierPro, WebHTTrack have been calculated as part of experiment.

An observation has shown that only 42.52 % of static links, 27.38 % of links generated by concatenation of strings and 15.18% of links generated by functions were retrieved by various open source crawlers. These results about types of links processed by various crawlers shows that crawlers cannot extract those links which were generated by complex methods due to using regular expressions, trying to discover new URLs by processing the code as text, instead of using simple interpreters and/or decompilers.

International Journal of Scientific & Engineering Research, Volume 6, Issue 9, September-2015
ISSN 2229-5518

Table 4. shows the summary of result of open source crawlers.

For Four methods Simple Average, Maximum Level, Weighted Average and Eight Level data has been collected about scale and assessment level of various open source crawlers. WebCopier, Heritrix get the best result in simple Average, Weighted average and Eight Levels methods. While in weighted Average WebCopier gives the best result which is followed by Heritrix.

Table 5. shows the results of classifying the crawling system according to scale and the assessment levels proposed by V.M.Preito et al.

4. Conclusion

A study on various open source crawlers concluded that all the available crawlers have their own advantage as well as consequences. One can use crawler according to their requirement. Scrapy is faster but it is not as scalable as Heritrix and Nutch. Scrapy is also an excellent choice for focused crawls. Heritrix is scalable but not dynamically scalable. Heritrix performs well in distributed environment. Nutch is highly scalable and also dynamically scalable through Hadoop. WIRE crawler provides good scalability. WebCopier, Heritrix gives the best result in terms of proposed scale. Various studies and experiment mentioned in this research paper highlights the various facts about different crawlers which will help the users to select crawler which will satisfy their needs.

TABLE 4
SUMMARY OF RESULTS OF OPEN SOURCE CRAWLERS [9]

	Heritrix	Nutch	Pavuk	Teleport	Web2disk	WebCopierPro	WebHTTrack
Text Link	2-100%	2-100%	2-100%	2-100%	2-100%	2-100%	2-100%
Href=Java Script Link	6-50%	3-25%	0-0%	4-33%	5-42%	12-100%	3-25%
Document write Link	6-50%	3-25%	0-0%	3-25%	4-33%	12-100%	2-17%
Menu Link	6-50%	3-25%	0-0%	3-25%	4-33%	12-100%	2-17%
Flash Link	0-0%	0-0%	0-0%	0-0%	0-0%	0-0%	2-50%
Applet Link	2-50%	0-0%	0-0%	0-0%	0-0%	0-0%	2-50%
Redirects	6-60%	6-60%	2-20%	6-60%	4-40%	0-0%	6-60%
Class or Java Link	0-0%	0-0%	0-0%	0-0%	0-0%	2-50%	0-0%
Ajax Link	0-0%	1-50%	0-0%	0-0%	0-0%	0-0%	0-0%
Links with a	2-50%	2-50%	0-0%	3-75%	2-50%	0-0%	2-50%
VbScript Link	3-75%	3-75%	0-0%	3-75%	3-75%	0-0%	2-50%
Static String link	26-62%	19-45%	4-10%	8-43%	22-52%	16-38%	22-52%
Concatenated String Link	6-50%	2—16%	0-0%	1-8%	1-8%	12-100%	1-8%
Special Function String Link	1-6%	2-13%	0-0%	5-31%	1-6%	12-75%	0-0%
Tests Passed	33-47%	23-33%	4-6%	24-34%	24-34%	40-57%	23-32%

International Journal of Scientific & Engineering Research, Volume 6, Issue 9, September-2015
ISSN 2229-5518

TABLE 5
RESULTS ACCORDING TO PROPOSED SCALE [9]

	Heritrix	Nutch	Pavuk	Teleport	Web2disk	WebCopier	WebHTTrack
Simple Average	3,77	2,63	0,46	2,29	2,74	4,57	2,40
Maximum Level	6,00	7,00	3,00	4,00	5,00	7,00	6,00
Weighted Average	1,63	0,99	0,14	0,75	1,03	3,42	0,86
Eight Levels	6,00	4,30	1,20	4,00	4,55	4,55	3,40

REFERENCES

[1] Quora, "What is the best open source crawler and why," http://www.quora.com/What-is-the-best-open-source-web-crawler-and-why.2010.
[2] BLIKK BLOG, "Comparison of open source web crawler,"http://blog.blikk.co/comparison-of-open-source-web-crawlers/.2014.
[3] Christopher Olston and Marc Najork, "Foundations and TrendsR in Information Retrieval", pp. 201.
[4] Baiju NT, Big Data Made Simple, http://bigdata-madesimple.com/top-50-open-source-web-crawlers-for-data-mining/.2015.
[5] Carlos Castillo, "EffectiveWeb Crawling", PhD dissertation, Dept. of Computer Science, University of Chile,2004.
[6] Junghoo Cho, Hector Garcia-Molina, "Parallel Crawler," unpublished.
[7] Vik's Blog, "A Comparison of Open Source Search Engines," https://zooie.wordpress.com/2009/07/06/a-comparison-of-open-source-search-engines-and-indexing-twitter/.2009.
[8] K.F. Bharati, Prof. P. Premchand and Prof. A Govardhan, "HIGWGET-A Model forCrawling Secure Hidden WebPages," International Journal of Data Mining & Knowledge Management Process ,Vol.3, No. 2, March 2013.
[9] Juan M. Corchado Rodríguez, Javier Bajo Pérez, Paulina Golinska, Sylvain Giroux, Rafael Corchuelo, "Trends in Practical Applications of Agents and Multiagent Systems", Springer Heidelberg New York Dordrecht London. Pp. 146,147.
[10] Andre Ricardo and Carlos Serrao, "Comparison of existing open source tools for web crawling and indexing of free music," Journal of Telecommunications Vol. 18, Issue 1, 2013.
[11] Christian Middleton, Ricardo Baeza-Yates, "A Comparison of Open Source Search Engines ," unpublished.
[12] Paolo Boldi, Bruno Codenotti, Massimo Santini, and Sebastiano Vigna. Ubicrawler: a scalable fully distributed web crawler. Software, Practice and Experience, 34(8):711–726, 2004.
[13] Junghoo Cho, Hector Garcia-Molina, and Lawrence Page., "Efficient crawling through url ordering," In Proceedings of the seventh conference on World Wide Web, Brisbane, Australia, 1998. Elsevier Science.
[14] Junghoo Cho and Hector Garcia-Molina, "Parallel crawlers," In Proceedings of the eleventh international conference on World Wide Web, Honolulu, Hawaii, USA, 2002. ACM Press.
[15] Chau, D. H., Pandit, S., Wang, S., and Faloutsos, C. 2007. Parallel crawling for online social networks. In Proceedings of the 16th international Conference on World Wide Web (Banff, Alberta, Canada, May 08 - 12, 2007). WWW '07. ACM, New York, NY, 1283-1284.
[16] http://nutch.apache.org/
[17] http://scrapy.org/
[18] https://webarchive.jira.com/wiki/display/Heritrix/Heritrix
[19] https://www.gnu.org/software/wget/
[20] http://www.pavuk.org/man.html
[21] http://sourceforge.net/
[22] http://www.cs.cmu.edu/~rcm/websphinx/
[23] http://j-spider.sourceforge.net/
[24] http://www.web2disk.com/
[25] http://www.tenmax.com/teleport/pro/home.htm
[26] https://www.httrack.com/