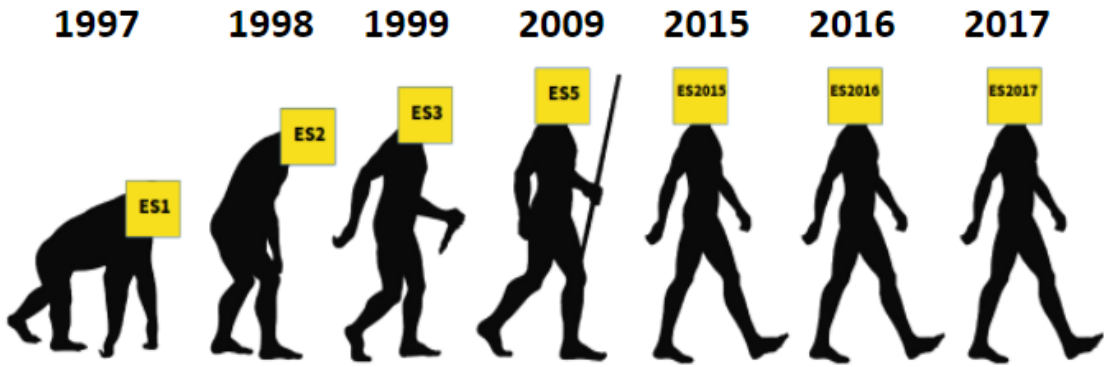
Node.js Document Store for Web Crawling

David Bui

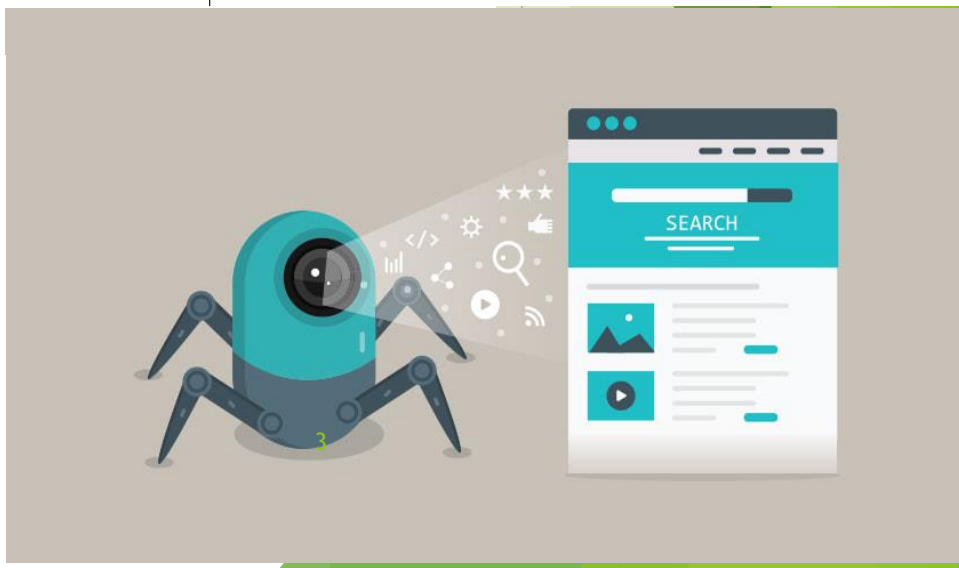
WARC-KIT



- A JavaScript tool kit for WARC files created in Node.js
- Comes with a WARC parsing tool known as WARCFilter
- Comes with a custom JavaScript only Linear Hash Table document store
- Overall feature is to create custom indices on WARC file collections

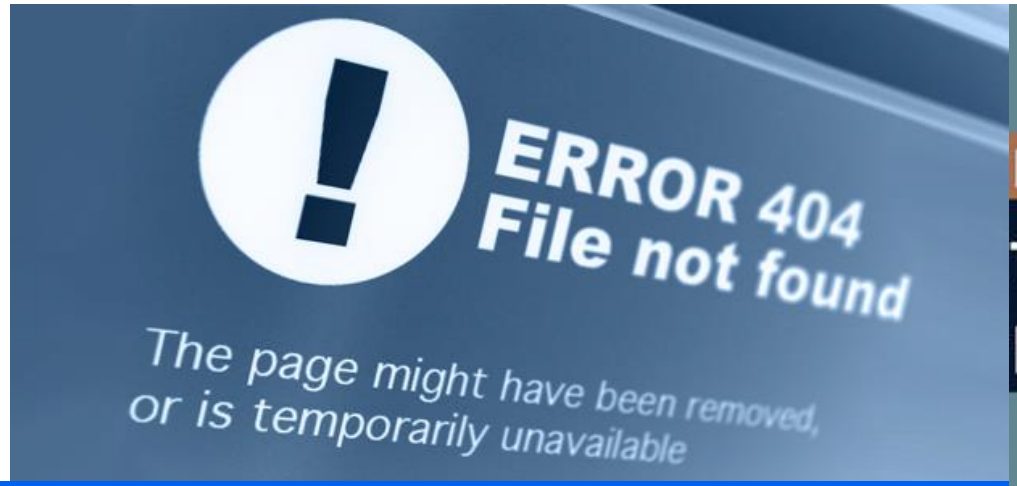
Web Crawling

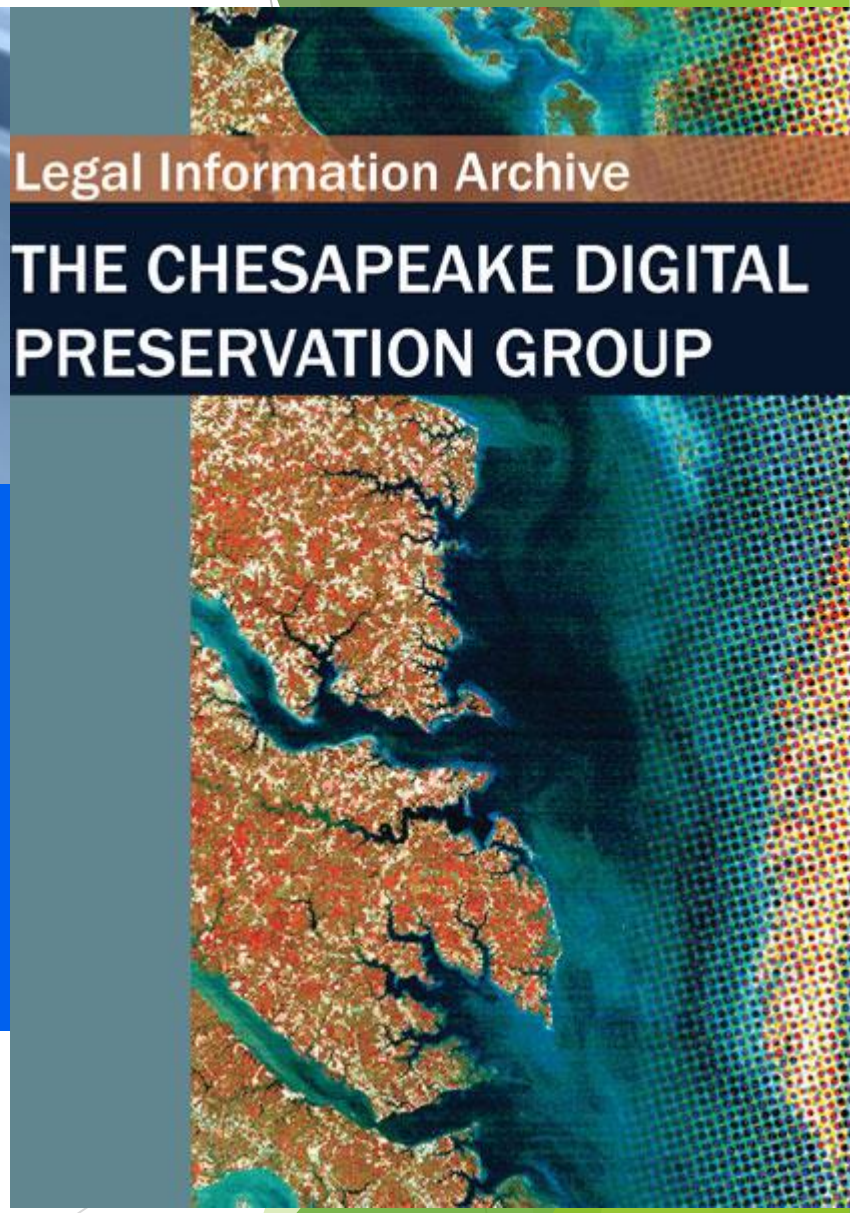
- Web Crawling: Programmatically browsing the internet through bots
  1. Search engine Indexing
  2. Web Archiving



Link Rot

- Unreachable web pages
- Dead links
- Deregistered domains
- Chesapeake Digital Preservation Group





Web Archiving

- Digital preservation for posterity.
- Commonly stored resources include web page content, images and videos

Archive File Formats

- ARC file format
- WARC file format

version-block == filedesc://<path><sp><version specific data><sp><length><r
<version-number><sp><reserved><sp><origin-code><nl>
<URL-record-definition><nl>
<nl>

version-1-block == filedesc://<path><sp><ip_address><sp><date><sp>text/pl
1<sp><reserved><sp><origin-code><nl>
<URL IP-address ArchivArchivee-date Content-type Archive-length<nl>
<nl>

version-2-block == filedesc://<path><sp><ip_address><sp><date><sp>text/pl
-<sp>-<sp>0<sp><filename><sp><length><nl>

2<sp><reserved><sp><origin-code><nl>
URL<sp>IP-address<sp>Archive-date<sp>Content-type<sp>Result-code<sp>
Offset<sp>Filename<sp>Archive-length<nl>

```
WARC/1.0
WARC-Type: response
WARC-Target-URI: http://www.archive.org/images/logoc.jpg
WARC-Date: 2006-09-19T17:20:24Z
WARC-Block-Digest: sha1:2ASS7ZUZY6ND6CCHXETFVJDENAWF7KQ2
WARC-Payload-Digest: sha1:CCHXETFVJD2MUZY6ND6SS7ZENMWF7KQ2
WARC-IP-Address: 207.241.233.58
WARC-Record-ID: <urn:uuid:39509228-ae2f-11b2-763a-aa4c6ec90bb0>
WARC-Segment-Number: 1
Content-Type: application/http;msgtype=response
Content-Length: 1600

HTTP/1.1 200 OK
Date: Tue, 19 Sep 2006 17:18:40 GMT
Server: Apache/2.0.54 (Ubuntu)
Last-Modified: Mon, 16 Jun 2003 22:28:51 GMT
ETag: "3e45-67e-2ed02ec0"
Accept-Ranges: bytes
Content-Length: 1662
Connection: close
Content-Type: image/jpeg

[first 1360 bytes of image/jpeg binary data here]
```

Current Tools

- warc: A Internet Archive Python WARC library (brief instructions)
- ia-hadoop-tools: A Internet Archive Java/Hadoop/Pig WARC tool (no documentation)
- webarchive-commons: Java WARC tools maintained by the IIPC
- warcit: Python library for converting html files to WARC files
- WARCIO: Python library for streaming WARC records.

Where’s everything else?

Java™

Node.js

- Language of the web
- Backend JavaScript runtime environment
- Simple, Fast, and Lightweight
- Node Package Manager(NPM) is awesome
- Shown to be 20 times faster than Ruby on Rails
- Few WARC related modules
- Let’s fix that

WARC File Format

- Extension of the Internet Archive's ARC File Format. Hence the name Web ARChive.
- The WARC file consists of a concatenation of one or more WARC records.
- There are 8 types of WARC records seen as seen to the left.
- A WARC record consists of
  - The header
  - Then Record content block
- The header has mandatory named fields
  - Date
  - Type
  - Length of the record
  - Plus, other fields that assist in retrieval
- The content block contains resources in any format such as images or audio

WARC Record Types

- ★ warcinfo
- ★ response
- ★ resource
- ★ request
- ★ metadata
- ★ revisit
- ★ conversion
- ★ continuation

```
WARC-Type   = "WARC-Type" ":" record-type
record-type = "warcinfo" | "response" | "resource"
            | "request" | "metadata" | "revisit"
            | "conversion" | "continuation"
```

http://iipc.github.io/warc-specifications/specifications/warc-format/warc-1.1/

@ibnesayeed 6

warcinfo record

```
WARC/1.0
WARC-Type: warcinfo
WARC-Date: 2011-02-25T18:32:19Z
WARC-Filename: WIDE-20110225183219005-04371-13730~crawl301.us.archive.org~9443.warc.gz
WARC-Record-ID: <urn:uuid:88fbcbee-f24e-47c1-b0c4-f7a9530ceb74>
Content-Type: application/warc-fields
Content-Length: 442

software: Heritrix/3.0.1-SNAPSHOT-20110127.213729 http://crawler.archive.org
ip: 207.241.232.79
hostname: crawl301.us.archive.org
format: WARC File Format 1.0
conformsTo: http://bibnum.bnf.fr/WARC/WARC_ISO_28500_version1_latestdraft.pdf
operator: kenji@archive.org
isPartOf: wide
description: seeds.txt
robots: obey
http-header-user-agent: Mozilla/5.0 (compatible; archive.org_bot +http://www.archive.org/details/archive.org_bot)
```

Response record with html content

```
WARC/1.0
WARC-Type: response
WARC-Target-URI: http://ricardo.parente.us/tags/ireland/
WARC-Date: 2011-02-25T18:33:10Z
WARC-Payload-Digest: sha1:7GH2LBI4Q65JIUOMVNWNASDO4DTPNK42
WARC-IP-Address: 208.205.181.39
WARC-Record-ID: <urn:uuid:db3a40a5-a313-40db-b63d-5406246d1ca3>
Content-Type: application/http; msgtype=response
Content-Length: 78688

HTTP/1.1 200 OK
Date: Fri, 25 Feb 2011 18:33:03 GMT
Server: Apache
X-Powered-By: PHP/5.2.16
X-Pingback: http://ricardo.parente.us/xmlrpc.php
Set-Cookie: PHPSESSID=26e4aa448afbaa96a525926fd7286028; path=/
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >

<head profile="http://gmpg.org/xfn/11">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Ireland &laquo;  ColdFusion Developers Network</title>

<link rel="alternate" type="application/rss+xml" title="ColdFusion Developers Network RSS Feed" href="http://ricardo.parente.us/feed/" />
<link rel="alternate" type="application/atom+xml" title="ColdFusion Developers Network Atom Feed" href="http://ricardo.parente.us/feed/atom/" />
<link rel="pingback" href="http://ricardo.parente.us/xmlrpc.php" />
<link rel="shortcut icon" href="http://ricardo.parente.us/wp-content/themes/arclite/favicon.ico" />
<script language="javascript" type="text/javascript"> if (top.location != location) {   top.location.href = document.location.href ;  }
</script>
```

CDX File

- Crawl Index (CDX) files consist of individual lines of text that each summarize a WARC record.
- Starts with a CDX legend that describes how each line of data is formatted.
- Used to index WARC files.

```
 CDX N b a m s k r M S V g
10,100,196,202)/musewebmain/dzyy/default.asp?ntmpkzh=236900 20110225232158 http://202.196.100.10/musewebmain/dzyy/default.asp?nTmpKzh=236900 text/html 200
4CVVD6FMLZCW73EVBVHVKBU4SPPOI3OE - - 587 824729713 testWARCfiles/WIDE-20110225221304846-04388-13730~crawl301.us.archive.org~9443.warc.gz
10,100,196,202)/musewebmain/dzyy/default.asp?ntmpkzh=265878 20110225190300 http://202.196.100.10/musewebmain/dzyy/default.asp?nTmpKzh=265878 text/html 200
4CVVD6FMLZCW73EVBVHVKBU4SPPOI3OE - - 587 231703167 testWARCfiles/WIDE-20110225184020081-04372-13730~crawl301.us.archive.org~9443.warc.gz
10,100,196,202)/musewebmain/dzyy/default.asp?ntmpkzh=265880 20110225190427 http://202.196.100.10/musewebmain/dzyy/default.asp?nTmpKzh=265880 text/html 200
4CVVD6FMLZCW73EVBVHVKBU4SPPOI3OE - - 587 243878729 testWARCfiles/WIDE-20110225184020081-04372-13730~crawl301.us.archive.org~9443.warc.gz
10,100,196,202)/musewebmain/dzyy/default.asp?ntmpkzh=266525 20110225232344 http://202.196.100.10/musewebmain/dzyy/default.asp?nTmpKzh=266525 text/html 200
4CVVD6FMLZCW73EVBVHVKBU4SPPOI3OE - - 586 987899236 testWARCfiles/WIDE-20110225220702321-04387-13730~crawl301.us.archive.org~9443.warc.gz
10,100,196,202)/musewebmain/dzyy/default.asp?ntmpkzh=266527 20110225232226 http://202.196.100.10/musewebmain/dzyy/default.asp?nTmpKzh=266527 text/html 200
4CVVD6FMLZCW73EVBVHVKBU4SPPOI3OE - - 586 828862927 testWARCfiles/WIDE-20110225221304846-04388-13730~crawl301.us.archive.org~9443.warc.gz
10,100,196,202)/musewebmain/dzyy/default.asp?ntmpkzh=600181887 20110225224003 http://202.196.100.10/musewebmain/dzyy/default.asp?nTmpKzh=600181887 text/
html 200 4CVVD6FMLZCW73EVBVHVKBU4SPPOI3OE - - 590 463925916 testWARCfiles/WIDE-20110225215415804-04385-13730~crawl301.us.archive.org~9443.warc.gz
```

WAT and WET

- Web Archive Transformation (WAT): JSON transformed WARC records
- WARC Encapsulated Text (WET): Plain Text only WARC record

```
Envelope
  WARC-Header-Metadata
    WARC-Target-URI [string]
    WARC-Type [string]
    WARC-Date [datetime string]
    ...
  Payload-Metadata
    HTTP-Response-Metadata
      Headers
        Content-Language
        Content-Encoding
        ...
      HTML-Metadata
        Head
          Title [string]
          Link [list]
          Metas [list]
        Links [list]
      Headers-Length [int]
      Entity-Length [int]
      ...
    ...
  ...
Container
  Gzip-Metadata [object]
  Compressed [boolean]
  Offset [int]
```

```
WARC/1.0
WARC-Type: conversion
WARC-Target-URI: http://news.bbc.co.uk/2/hi/africa/3414345.stm
WARC-Date: 2014-08-02T09:52:13Z
WARC-Record-ID:
WARC-Refers-To:
WARC-Block-Digest: sha1:JROHLCS5SKMBR6XY46WXREW7RXM64EJC
Content-Type: text/plain
Content-Length: 6724

BBC NEWS | Africa | Namibia braces for Nujoma exit
...
President Sam Nujoma works in very pleasant surroundings in the small but beautif
ul old State House...
```

WARCFilter

- CLI program to parse and filter out WARC records
- Create new WARC files using records from existing collections
- Create CDX index files on WARC files
- Parse CDX files and retrieve WARC records.
- Create Webgraph datasets from Common Crawl's dataset

Arguments format and cli directions

If everything is running this cli interface should appear in console

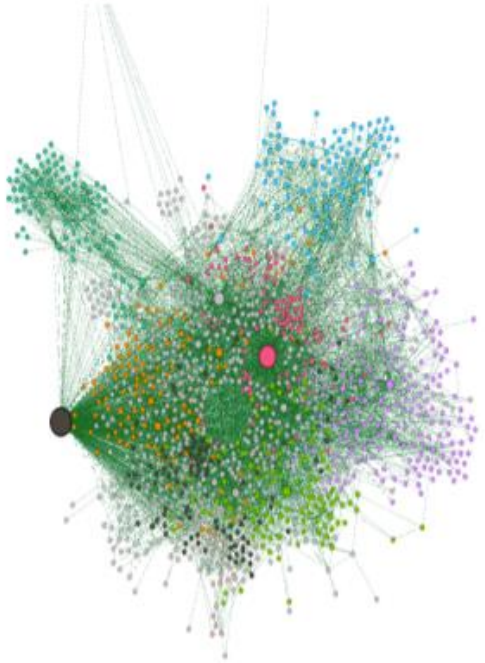
```
Enter in this format: src: origFile dest: destinationFile mode: mode {arguments}, press e to exit:
```

Below is a list of possible arguments

- src: {comma separated list path to files to read from}
- dest: {path of file to write warc records to}
- mode: {warc | cdx | createCDX, genCCWebGraph}
- type: {cdx | cdxj} (only used in createCDX mode)
- url: {comma separated list of urls}
- fileType: {comma separated list of file types}
- date: {yyyymmddhhmmss | yyyymmddhhmmss - yyyymmddhhmmss} (ranged queries accepted inclusive - exclusive)
- recordLimit: {int} (limit of number of records to write from filter)
- watLimit: (int) (genCCWebgraph mode only limits the number of WAT files read)
- pathOffset: (int) (genCCWebgraph mode only offset into a Common Crawl path file to read)

```
PS D:\Desktop\WARCFilter> npm start

> warcparser@1.0.0 start
> nodemon index.js

[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Enter in this format: origFile, destination file, {arguments}, t for a timing test or  press e to exit:
src: ./warc/example.warc.gz dest: ./tests/exampleCDXCreate.cdx mode: createCDX type: cdx
running cdx creator...
Creating your cdx at ./tests/exampleCDXCreate.cdx ...
Finished writing 42800  indexes total
24.533 seconds were needed to create the cdx file


Enter in this format: origFile, destination file, {arguments}, t for a timing test or  press e to exit:
```

Webgraph Dataset Creation



- Generated using a Common Crawl wat.paths file.
- Datasets are generated as a single compressed text file.
- Each line represents a directed edge.

```
http://17hmr.net/index.php?topic=9520.msg130865 http://validator.w3.org/check?uri=referer
http://2012indyinfo.com/2012/05/24/bbc-news-facebook-and-banks-behind-flotation-face-lawsuit/ http://www.bluehost.com/
http://2012indyinfo.com/2012/05/24/bbc-news-facebook-and-banks-behind-flotation-face-lawsuit/ http://www.bluehost.com/
http://2012indyinfo.com/2012/05/24/bbc-news-facebook-and-banks-behind-flotation-face-lawsuit/ http://www.bluehost.com/cgi/help
http://2012indyinfo.com/2012/05/24/bbc-news-facebook-and-banks-behind-flotation-face-lawsuit/ http://www.bluehost.com/cgi/info/contact_us
http://2012indyinfo.com/2012/05/24/bbc-news-facebook-and-banks-behind-flotation-face-lawsuit/ http://www.bluehost.com/cgi/info/about_us
http://2012indyinfo.com/2012/05/24/bbc-news-facebook-and-banks-behind-flotation-face-lawsuit/ http://www.bluehost.com/cgi-bin/partner
http://2012indyinfo.com/2012/05/24/bbc-news-facebook-and-banks-behind-flotation-face-lawsuit/ http://www.bluehost.com/cgi/terms
http://247magazine.co.uk/2011/07/19/review-2000-trees-festival-2011/ https://twitter.com/247magazine
http://247magazine.co.uk/2011/07/19/review-2000-trees-festival-2011/ https://www.facebook.com/pages/247-Magazine/6541655414
http://247magazine.co.uk/2011/07/19/review-2000-trees-festival-2011/ http://247magazine.skiddletickets.com/events.php
http://247magazine.co.uk/2011/07/19/review-2000-trees-festival-2011/ https://twitter.com/share
http://247magazine.co.uk/2011/07/19/review-2000-trees-festival-2011/ http://1.gravatar.com/avatar/7f1b70e3fee8e2095dd3891a68a92772?s=70&amp;d=http%3A%2F%2F1.gravatar.com%2Favatar%2Fad516503
http://247magazine.co.uk/2011/07/19/review-2000-trees-festival-2011/ http://www.muzu.tv/channel/247magazine/playlists/247-magazine-music-videos/1194942/
http://247magazine.co.uk/2011/07/19/review-2000-trees-festival-2011/ http://www.muzu.tv/
http://247magazine.co.uk/2011/07/19/review-2000-trees-festival-2011/ http://wordpress.org/
http://247magazine.co.uk/2011/07/19/review-2000-trees-festival-2011/ http://www.gabfirethemes.com/
http://247wallst.com/investing/2013/02/06/the-top-dividend-yields-from-the-bofamerrill-lynch-model-portfolio-changes/ http://b.scorecardresearch.com/p?c1=2&c2=16807273&cv=2.0&cj=1
http://247wallst.com/investing/2013/02/06/the-top-dividend-yields-from-the-bofamerrill-lynch-model-portfolio-changes/ https://s0.wp.com/wp-content/themes/vip/247wallst/images/search-icon.pn
http://247wallst.com/investing/2013/02/06/the-top-dividend-yields-from-the-bofamerrill-lynch-model-portfolio-changes/ http://www.magnetmail.net/actions/subscription_form_action_24new.cfm
http://247wallst.com/investing/2013/02/06/the-top-dividend-yields-from-the-bofamerrill-lynch-model-portfolio-changes/ https://s0.wp.com/wp-content/themes/vip/247wallst/images/social_icons/F
http://247wallst.com/investing/2013/02/06/the-top-dividend-yields-from-the-bofamerrill-lynch-model-portfolio-changes/ http://www.facebook.com/247WallSt
http://247wallst.com/investing/2013/02/06/the-top-dividend-yields-from-the-bofamerrill-lynch-model-portfolio-changes/ https://s0.wp.com/wp-content/themes/vip/247wallst/images/social_icons/T
http://247wallst.com/investing/2013/02/06/the-top-dividend-yields-from-the-bofamerrill-lynch-model-portfolio-changes/ http://twitter.com/247wallst
http://247wallst.com/investing/2013/02/06/the-top-dividend-yields-from-the-bofamerrill-lynch-model-portfolio-changes/ https://plus.google.com/109889536671975286106?prsrc=3
http://247wallst.com/investing/2013/02/06/the-top-dividend-yields-from-the-bofamerrill-lynch-model-portfolio-changes/ https://s0.wp.com/wp-content/themes/vip/247wallst/images/menu/rss.png
http://247wallst.com/investing/2013/02/06/the-top-dividend-yields-from-the-bofamerrill-lynch-model-portfolio-changes/ http://feeds.feedburner.com/typepad/RyNm
http://247wallst.com/investing/2013/02/06/the-top-dividend-yields-from-the-bofamerrill-lynch-model-portfolio-changes/ https://twitter.com/share
http://247wallst.com/investing/2013/02/06/the-top-dividend-yields-from-the-bofamerrill-lynch-model-portfolio-changes/ http://247wallst.dailyfinance.com/quote/nyse/altria-group-inc/mo
```

JavaScript Databases

lowdb downloads 1.6M/month Node.js CI passing

- In memory ones do exist
- On disk databases are nonexistent
- Hybrid On disk databases are plentiful.
- Database storage structure: B+ tree, Log Structured Merge tree, Hash Tables
- URL Key -> WARC record

HarperDB™

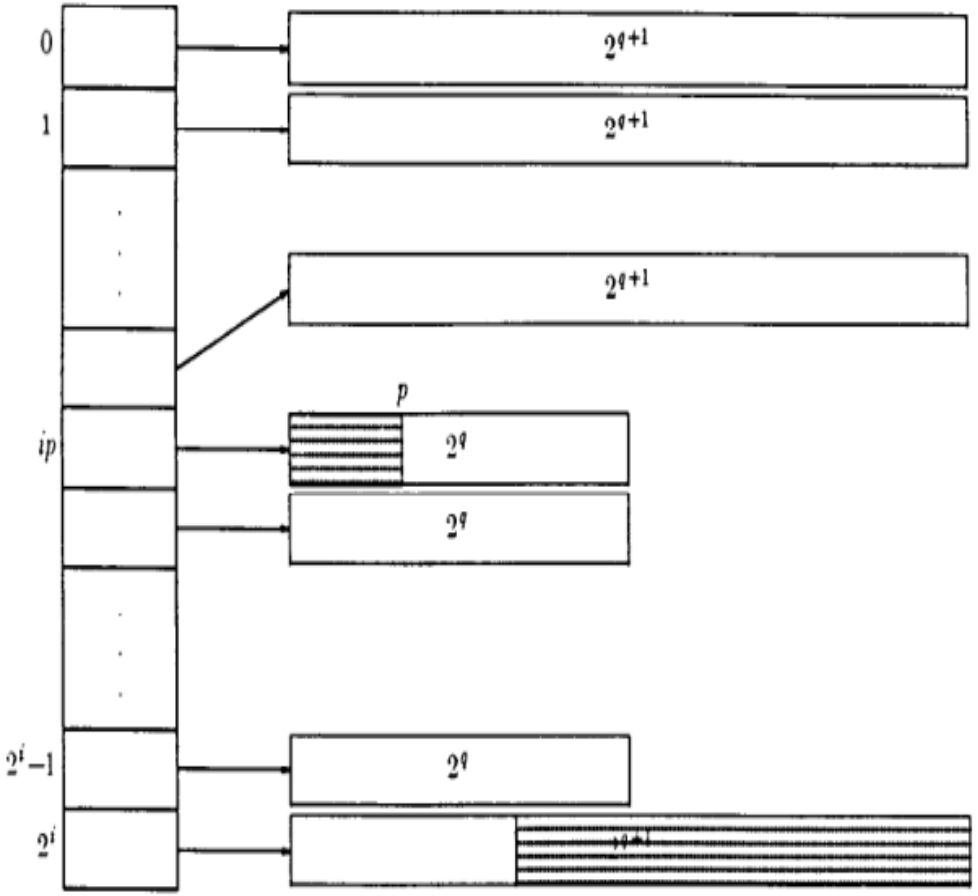
levelDB

LMDB

RocksDB

Linear Hashing Explained

- A hash function will give typically give some number of bits. Let's say our hash function gives 32-bit output from some key. However, in Linear Hashing we will only use the first I bits since we only start with N buckets.
- If we start with N= 2 buckets, then I = 1 bits. So, we will only use the first bit of the hash function's 32-bit output to map to a bucket.
- Once number of insertions exceed the load factor add 1 bucket to N. If N >(2^I -1) we need to increment I to address to the new bucket.
- When any bucket is added we split the bucket at index S's keys with the new bucket, rehash if I is incremented, and then increment S. Once N has doubled from where it was initially, we reset S to 0.



Linear Hash Table Implementation

- Have folders representing a bucket.
- Have header .hix files with offset and length of a .txt data file.
- Implemented put, get, delete, and update functions.
- Use Streams to maximize speed.
- Key-value is limited documented oriented is better

3.hix - Notepad

File Edit Format View Help

```
7206419495114264579,0,9
8544993663200139267,9,9
13915196356442311683,18,9
9996227108599564291,27,9
15120555784665207811,36,9
12659576498202810371,45,9
5581953373794281475,54,10
```

3.txt - Notepad

File Edit Format View Help

```
2
Gfr,france3,ce-soir-ou-jamais)/img/jpg/cardinal.jpgimage/jpeg20110225183251619639962 22WIDE-201102251832
Gjp,co,akibaoo)/new/upimg/m_3963.jpgimage/jpeg20110225183333995379555 41WIDE-20110225183219005-04371-137
183219005-04371-13730_crawl301.us.archive.org_9443.warc.gzS     Gar,com,adsclasificados,campogallo)/q/c
Gcom,198ic,t1091nl)/stock-product/images/r217/r217016-01.jpgimage/jpeg20110225184137147599852765WIDE-20
/science/photos/photo_science.php?gallery_vignvcmid=715ee04f3e2b4110vgnvcm100000ee02a8c0rcrd&no=8text/htm
        Gcom,kenanaonline,media)/photos/1238068/1238068892/thumbnail_1238068892.jpg?1296099593=image/jpeg
        Gcom,louboutinshoesale)/bmz_cache/4/48cc855249be579f2c25671142715c6c.image.33x50.jpgimage/jpeg201
3       Gcn,com,anheng,tls2200)/products/showimg.php?iid=828image/pjpeg20110225185828546222796797 6WIDE-20
        Gcom,wkbn)/media/lib/53/7/e/5/7e5240e3-30c6-4550-977b-a185cb319bbc/headline.jpgimage/jpeg20110225
        Gcom,subirimagenes,s2)/otros/previo/thump_38739481.jpgimage/jpeg20110225190023130732510896 51WIDE-
atext/html20110225190450463229637148 2WIDE-20110225183219005-04371-13730_crawl301.us.archive.org_9443.warc
        Gcn,fskm)/uploadfile/2009/1124/20091124051856609.jpgimage/jpeg20110225190606413930928414 4WIDE-201
        Gedu,salem)/athletics/salem_athletic_logo_5.jpg/resolveuid/0af4f60ac226b8749fbb4125b42ab6be/image
022519082810485353867617WIDE-20110225183219005-04371-13730_crawl301.us.archive.org_9443.warc.gz8
        Ginfo,anapakurort)/forum/userpix/26_200908_27.jpgimage/jpeg20110225190937100015370220222WIDE-2011
ntent/news/pastate/story/w-pa-police-searching-for-hatchet-wielding-mummy/4rloqwkl3uyc6s0o_kfizg.cspxtext
ive.org_9443.warc.gzX       Gar,com,adsclasificados,monteros)/q/servicios-sid85-cid7/page14/scort%20m
q/cat-cid2/page3/venta%20de%20le&text/html20110225191914836247909165 5WIDE-20110225183219005-04371-13730_c
        Gru,superkover)/content/images/water/76ee3de97a1b8b903319b7c013d8c877/386_600/1271071397_image.jp
        Gar,com,adsclasificados,resistencia)/publicacion/images/204256_1_small.jpgimage/jpeg2011022519245
        Gcom,rey-estates)/uploads/img/auto/imo/104_peq_104_alvaro_siza_vieria_1_14_gde_1182422935.jpgimag
atalog%2fhtml%2fgraduate%2520catalog%2f44.csstext/html20110225193641225991983343 7WIDE-20110225183219005-0
.gz@        Gru,7347,aznakaevo)/post_info.php?category_id=325&sel_city_id=176text/html20110225193827 4
archive.org_9443.warc.gzW       Gorg,aiaa,ebooks)/bookstore/pagedisplay.do?genre=book&id=9781600 8
```

Pack

- Originally a Perl function
- Encode primitive variables into a binary String
- boolean: 1 byte, short 2 bytes, int 4 bytes, doubles 8 bytes,
- Has an unpack function to reverse the process.

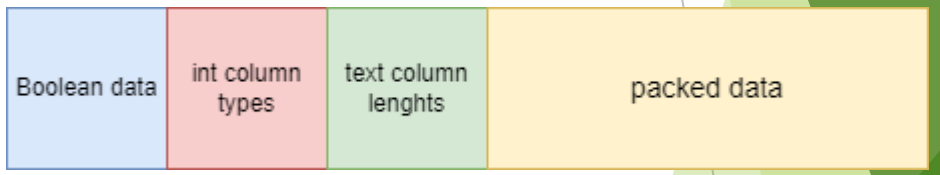
#	code	expected result
1	`pack('nvc*', 0x1234, 0x5678, 65, 66)`	`'\u00124xVAB'`
2	`pack('H4', '2345')`	`'#E'`
3	`pack('H*', 'D5')`	`'Õ'`
4	`pack('d', -100.876)`	`"\u0000\u0000\u0000\u0000\u00008YÀ"`

PackedTableTools

- Yioop's PackedTableTools
- JavaScript Port
- Define a table format for a set of records
- Packs an array of records into a String.
- Make Hash Table document oriented

```
let table_factory = new PackedTableTools(
    {"PRIMARY KEY" : "ID", "A": "DOUBLE", "B" : "TEXT",
     "C" :"TEXT", "D": "BOOL", "E" : "INT"});
```

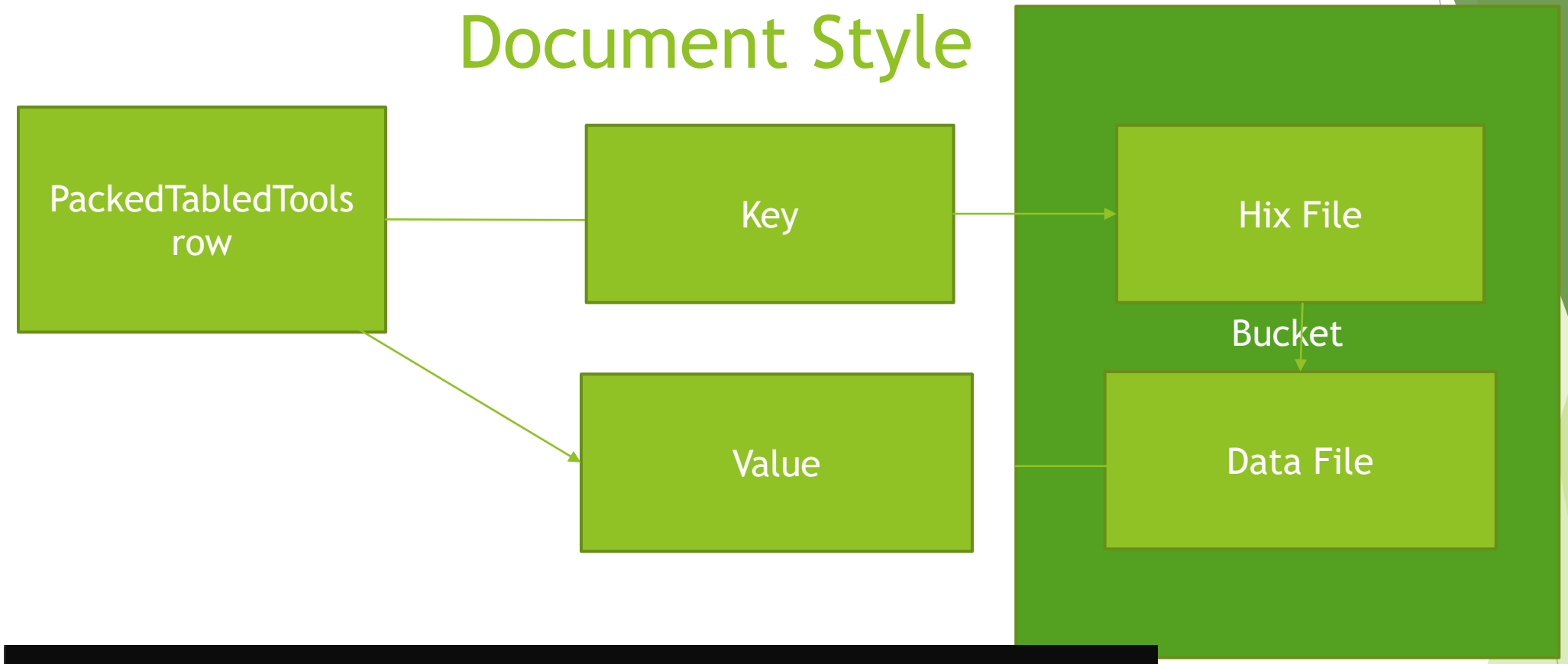
```
Original rows:  [
  { ID: 1, A: 1.59, B: 'Bong', C: 'yockledoo', D: true, E: 1000 },
  { ID: 2, A: 100.4, B: 'Blah', C: 'doodoo', D: false, E: 9990 }
]
```




```
Packed String:  ☻@♦    ⊖;Bongyockledoo♥è@♦♦d♦Blahdoodoo'♠
```

```
Unpacked:  [
  { A: 1.59, B: 'Bong', C: 'yockledoo', D: true, E: 1000 },
  { A: 100.4, B: 'Blah', C: 'doodoo', D: false, E: 9990 }
]
```

Document Style




```
Original Rows:  { ID: 1, A: 1.59, B: 'Bong', C: 'yockledoo', D: true, E: 1000 }
Packed string: ☺◘@♦     ☻;Bongyockledoo♥è
Unpacked: [ { A: 1.59, B: 'Bong', C: 'yockledoo', D: true, E: 1000 } ]


Original Rows:  { ID: 2, A: 100.4, B: 'Blah', C: 'doodoo', D: false, E: 9990 }
Packed string: ☺@♦♠d♦Blahdoodoo'♠
Unpacked: [ { A: 100.4, B: 'Blah', C: 'doodoo', D: false, E: 9990 } ]
```

Express.js

- All databases need an API
- De facto server framework for Node.js
- Common database operations implemented through HTTP routes

Express JS

GraphQL

- Query language for APIs
- Strictly Defined Schema
- Can be combined with Express.js
- Comes with Graph(i)QL a GUI for queries

```
type Query {
    greeting:String
    students:[Student]
}

type Student {
    id:ID!
    firstName:String
    lastName:String
    password:String
    collegeId:String
}
```

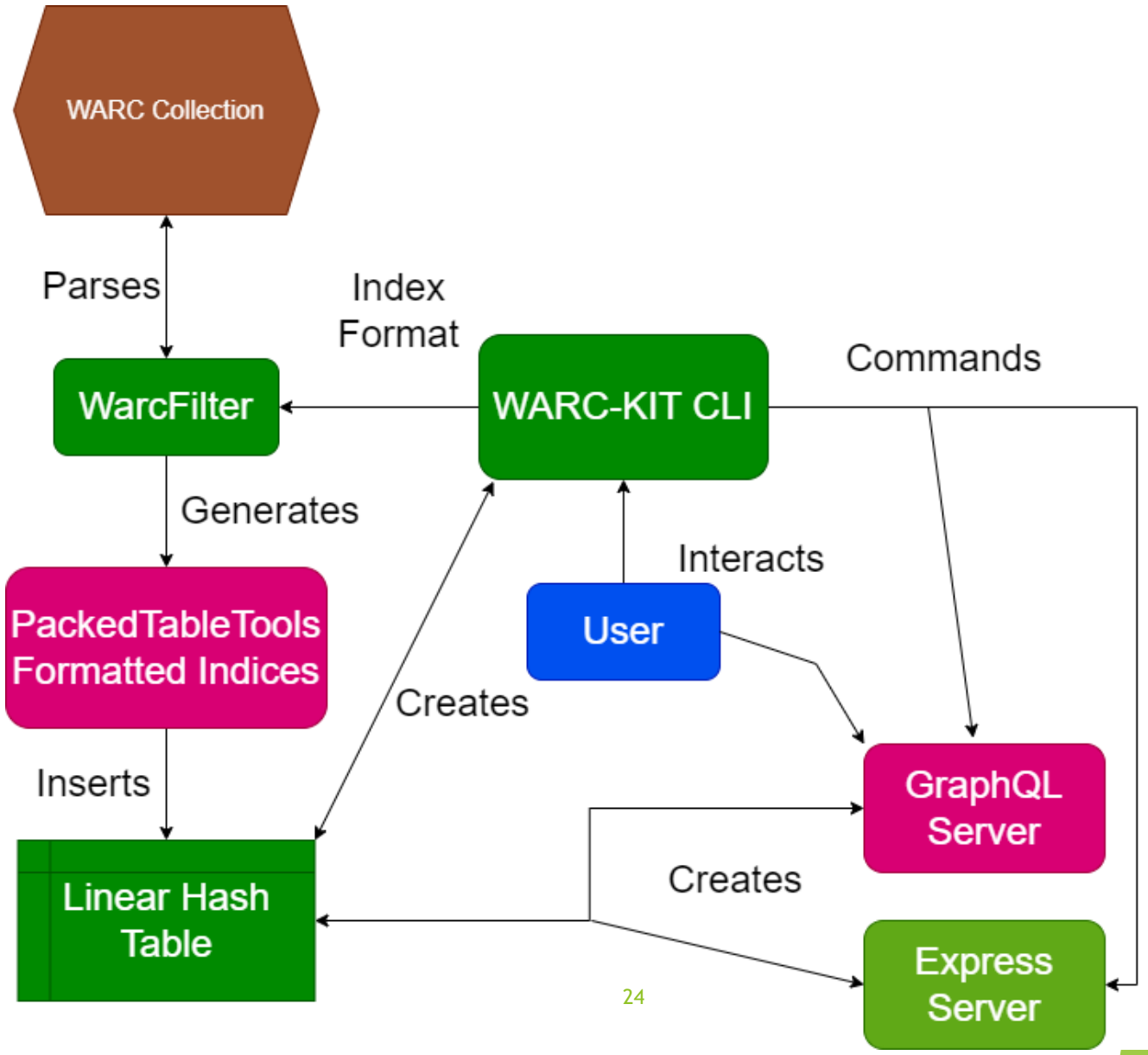
GraphiQL ▶ Prettify Merge Copy History

```
query JPG{
  queryIndex(fields: "MIME", values: "image/jpeg") {
    URLKEY
    MIME
    TIMESTAMP
    LENGTH
    OFFSET
    FILENAME
  }
}



# Welcome to GraphiQL
#
# GraphiQL is an in-browser tool for writing, validating, and
# testing GraphQL queries.
#
# Type queries into this side of the screen, and you will see intelligent
# typeaheads aware of the current GraphQL type schema and live syntax and
# validation errors highlighted within the text.
#
# GraphQL queries typically start with a "{" character. Lines that start
# with a # are ignored.
#
# An example GraphQL query might look like:
#
#     {
#       field(arg: "value") {
#         subField
#       }
#     }
#
# Keyboard shortcuts:
#
#  Prettify Query:  Shift-Ctrl-P (or press the prettify button above)
#
#     Merge Query:  Shift-Ctrl-M (or press the merge button above)
#
```


```
    },
    {
      "URLKEY": "ar,com,adsclasificados,burruyacu)/publicacion/images/266264_1_small.jpg",
      "MIME": "image/jpeg",
      "TIMESTAMP": "20110225183740",
      "LENGTH": "2241",
      "OFFSET": "78771057",
      "FILENAME": "WIDE-20110225183219005-04371-13730_crawl301.us.archive.org_9443.warc.gz"
    },
    {
      "URLKEY": "at,dvd-forum)/bilder/film_bilder/128782438223966700.jpg",
      "MIME": "image/jpeg",
      "TIMESTAMP": "20110225192333",
      "LENGTH": "90282",
      "OFFSET": "545134918",
      "FILENAME": "WIDE-20110225183219005-04371-13730_crawl301.us.archive.org_9443.warc.gz"
    },
    {
      "URLKEY": "net,myanimelist,cdn)/images/useravatars/81668.jpg",
      "MIME": "image/jpeg",
      "TIMESTAMP": "20110225194034",
      "LENGTH": "3582",
      "OFFSET": "968960686",
      "FILENAME": "WIDE-20110225183219005-04371-13730_crawl301.us.archive.org_9443.warc.gz"
    },
    {
      "URLKEY": "ar,com,adsclasificados,villasylvina)/publicacion/images/119409_1_small.jpg",
      "MIME": "image/jpeg",
      "TIMESTAMP": "20110225184827",
      "LENGTH": "2936",
      "OFFSET": "85809121",
      "FILENAME": "WIDE-20110225184020081-04372-13730_crawl301.us.archive.org_9443.warc.gz"
    },
    {
      "URLKEY": "ar,com,adsclasificados,puertorico)/publicacion/images/254585_1_small.jpg",
      "MIME": "image/jpeg",
      "TIMESTAMP": "20110225184427",
      "LENGTH": "2507",
      "OFFSET": "125760811",
      "FILENAME": "WIDE-20110225183219005-04371-13730_crawl301.us.archive.org_9443.warc.gz"
    }
```

WARC-KIT Functionality

- Create PackedTableTools format on a WARC file collection
- WARCFilter to parse WARC files and generate PackedTableTools Indices
- Insert Indices into Linear Hash Table.
- Create Express interaction server and GraphQL query server upon the Linear Hash Table.



WARC-KIT

List of WARC-KIT CLI commands

- create database {path} (is a folder that contains tables)
- create table {PackedTableTools formatted object}
- list databases {path} (shows current databases in path)
- list table {path} (shows current tables in path)
- use database {path} (set active database)
- use table {path} (set active table)
- delete database {path}
- delete table {path}
- create index {path} (runs the indicies creation operation on WARC files at {path} and inserts indices into current table's TableFormat)
- getIndex(key:String!): PackedTableFormat (GraphQL formatted function with returning the PackedTableTools formatted index corresponding to the argument key)
- query(fields: String, values: String): [PackedTableFormat] (GraphQL formatted query with fields and values arguments corresponding to the generated PackedTableFormat)

Country Crawl

- Find records with .country domains

```
const PACKED_INDEX = {"Primary Key": "URL", "COUNTRY":"TEXT", "MIME": "TEXT", "TIMESTAMP": "TEXT",
"LENGTH": "INT", "OFFSET": "INT", "FILENAME": "TEXT"};
```

```
query RUPNG {
  queryIndex(fields: "COUNTRY,MIME", values: "ru,png") {
    COUNTRY
    MIME
    TIMESTAMP
    LENGTH
    OFFSET
    FILENAME
  }
}
```


```
{
  "data": {
    "queryIndex": [
      {
        "COUNTRY": "ru",
        "MIME": "image/png",
        "TIMESTAMP": "20110225184849",
        "LENGTH": "1827",
        "OFFSET": "88347017",
        "FILENAME": "WIDE-20110225184020081-04372-13730_crawl301.us.archive.org_9443.warc.gz"
      },
      {
        "COUNTRY": "ru",
        "MIME": "image/png",
        "TIMESTAMP": "20110225185652",
        "LENGTH": "1954",
        "OFFSET": "216581392",
        "FILENAME": "WIDE-20110225183219005-04371-13730_crawl301.us.archive.org_9443.warc.gz"
      },
      {
        "COUNTRY": "ru",
        "MIME": "image/png",
        "TIMESTAMP": "20110225191812",
        "LENGTH": "23101",
        "OFFSET": "398482364",
        "FILENAME": "WIDE-20110225184020081-04372-13730_crawl301.us.archive.org_9443.warc.gz"
      },
      {
        "COUNTRY": "ru",
        "MIME": "image/png",
        "TIMESTAMP": "20110225190717",
        "LENGTH": "8379",
        "OFFSET": "270120545",
        "FILENAME": "WIDE-20110225184020081-04372-13730_crawl301.us.archive.org_9443.warc.gz"
      },
      {
        "COUNTRY": "ru",
        "MIME": "image/png",
        "TIMESTAMP": "20110225183746",
        "LENGTH": "750",
        "OFFSET": "79242129",
        "FILENAME": "WIDE-20110225183219005-04371-13730_crawl301.us.archive.org_9443.warc.gz"
      },
```

```
query DEPNG {
  queryIndex(fields: "COUNTRY,MIME", values: "de,png") {
    COUNTRY
    MIME
    TIMESTAMP
    LENGTH
    OFFSET
    FILENAME
  }
}
```

```
query USPNG {
  queryIndex(fields: "COUNTRY,MIME", values: "gov,jpeg") {
    COUNTRY
    MIME
    TIMESTAMP
    LENGTH
    OFFSET
    FILENAME
  }
}
```


```
{
  "data": {
    "queryIndex": [
      {
        "COUNTRY": "de",
        "MIME": "image/png",
        "TIMESTAMP": "20110225211320",
        "LENGTH": "1438",
        "OFFSET": "237861712",
        "FILENAME": "WIDE-20110225210142891-04382-13730_crawl301.us.archive.org_9443.warc.gz"
      },
      {
        "COUNTRY": "de",
        "MIME": "image/png",
        "TIMESTAMP": "20110225215159",
        "LENGTH": "26808",
        "OFFSET": "733293157",
        "FILENAME": "WIDE-20110225210142891-04382-13730_crawl301.us.archive.org_9443.warc.gz"
      },
      {
        "COUNTRY": "de",
        "MIME": "image/png",
        "TIMESTAMP": "20110225215413",
        "LENGTH": "2034",
        "OFFSET": "754016179",
        "FILENAME": "WIDE-20110225210142891-04382-13730_crawl301.us.archive.org_9443.warc.gz"
      },
      {
        "COUNTRY": "de",
        "MIME": "image/png",
        "TIMESTAMP": "20110225184800",
        "LENGTH": "2479",
        "OFFSET": "160631955",
        "FILENAME": "WIDE-20110225183219005-04371-13730_crawl301.us.archive.org_9443.warc.gz"
      },
      {
        "COUNTRY": "de",
        "MIME": "image/png",
        "TIMESTAMP": "20110225211122",
        "LENGTH": "1965",
        "OFFSET": "162286115",
        "FILENAME": "WIDE-20110225210142891-04382-13730_crawl301.us.archive.org_9443.warc.gz"
```

```
{
  "data": {
    "queryIndex": [
      {
        "COUNTRY": "gov",
        "MIME": "image/jpeg",
        "TIMESTAMP": "20110225214122",
        "LENGTH": "245269",
        "OFFSET": "634633711",
        "FILENAME": "WIDE-20110225210142891-04382-13730_crawl301.us.archive.org_9443.warc.gz"
      },
      {
        "COUNTRY": "gov",
        "MIME": "image/jpeg",
        "TIMESTAMP": "20110225190830",
        "LENGTH": "2548",
        "OFFSET": "354345908",
        "FILENAME": "WIDE-20110225183219005-04371-13730_crawl301.us.archive.org_9443.warc.gz"
      },
      {
        "COUNTRY": "gov",
        "MIME": "image/jpeg",
        "TIMESTAMP": "20110225191322",
        "LENGTH": "5245",
        "OFFSET": "346429869",
        "FILENAME": "WIDE-20110225184020081-04372-13730_crawl301.us.archive.org_9443.warc.gz"
      },
      {
        "COUNTRY": "gov",
        "MIME": "image/jpeg",
        "TIMESTAMP": "20110225192915",
        "LENGTH": "56184",
        "OFFSET": "686408358",
        "FILENAME": "WIDE-20110225183219005-04371-13730_crawl301.us.archive.org_9443.warc.gz"
      },
      {
        "COUNTRY": "gov",
        "MIME": "image/jpeg",
        "TIMESTAMP": "20110225200248",
        "LENGTH": "41798",
        "OFFSET": "969882399",
        "FILENAME": "WIDE-20110225184020081-04372-13730_crawl301.us.archive.org_9443.warc.gz"
      },
```

WARCFilter Experiments

- Dying Dell G7 15 Laptop: I7 8750H 16 GB ram Samsung 970 evo SSD
- Small Internet Archive WARC dataset
- CDX files vastly speed up filter time.
- Current JS tools for WARC provide only pure parsing.
- Web graph generation is resource intensive.

WARC File Name	Total # of records	WARC Filter Time	CDX Filter Time
WIDE-20110225183219005-04371	42800	38.716 sec.	0.938 sec
WIDE-20110225184020081-04372	57557	39.655 sec.	1.876 sec.
WIDE-20110225210142891-04382	43129	37.162 sec.	2.94 sec.
WIDE-20110225215415804-04385	44646	37.456 sec.	1.812 sec.
WIDE-20110225221304846-04388	50493	39.367 sec.	3.099 sec.

Table 4.1: Comparison of the time WARCFilter takes to parse and filter WARC files directly versus parsing their CDX files.

Table 4.3: Table showing the time and number of edges generated for each Common Crawl dataset paths file.

Common Crawl WAT datasets	# Edges Generated	Generation time
cc-nov-2015-wat.paths	188,721,679	4219.540 sec.
cc-dec-2016-wat.paths	120,007,747	4268.837 sec.
cc-nov-2017-wat.path	81,434,921	4085.955 sec.
cc-nov-2018-wat.path	75,781,930	4303.162 sec.
cc-nov-2019-wat.path-04385	81,100,750	4135.753 sec.
cc-nov-dec-2020-wat.path	57,240,145	3818.900 sec.

Table 4.2: Comparison of the time *node-warc* takes to parse through WARC file Records

WARC File Name	node-warc parse time
WIDE-20110225183219005-04371	14.187 sec.
WIDE-20110225184020081-04372	16.163 sec.
WIDE-20110225210142891-04382	14.859 sec.
WIDE-20110225215415804-04385	16.187 sec.
WIDE-20110225221304846-04388	14.919 sec.

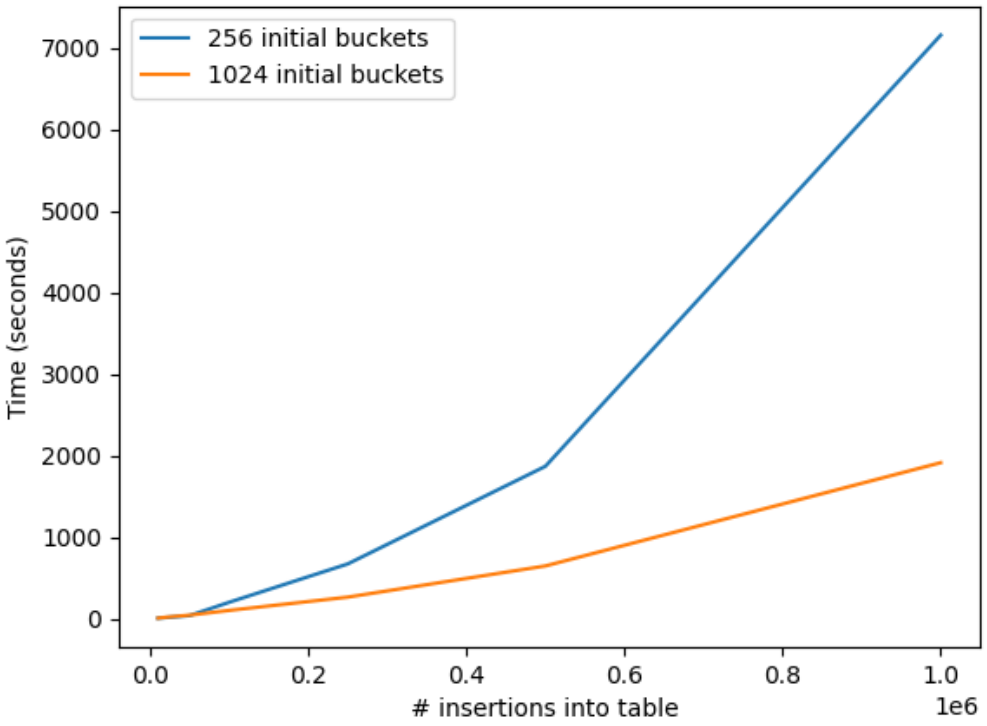


Table 4.4: Get performance test between our JavaScript implemented Linear Hash Table vs a Rust implemented Linear Hash Table

Number of gets	LHT256 time	LHT1024 time	Rust LHT time
10,000 key-value gets	2.023 sec.	2.634 sec.	8.582 sec.
100,000 key-value gets	42.733 sec.	40.267 sec.	102.321 sec.

Linear Hash Table Experiments

- Initial bucket configuration is crucial.
- Average 1,500 inserts/second
- Get tests on average are around 2,500 gets /second

WARC-KIT Experiments

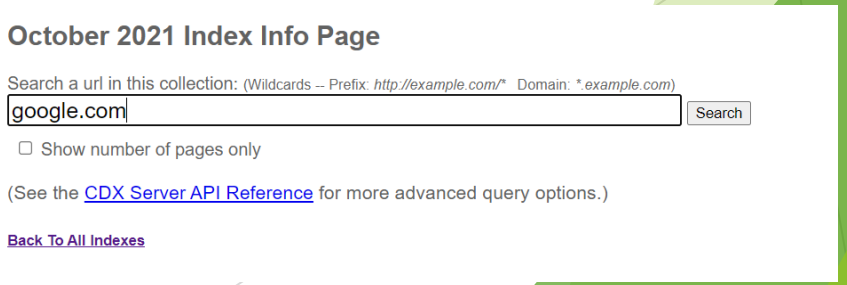
- Initial indices creation comparable to Linear Hash Table insert time.
- Average 46 seconds to create an index upon 1 WARC file.
- Queries by URL are instant, while complex queries take longer but are consistent.
- Common Crawl index server has similar functionality.

Table 4.5: Time to create and insert PackedTableTools formatted indices into Linear Hash Table and create a GraphQL query server on a WARC file collection

1 WARC file	3 WARC files	5 WARC files
39.741 sec.	156.639 sec.	238.414 sec.

Table 4.6: Various Queries on the GraphQL query server created from a indexed WARC file collection.

Query Name	1 WARC file index	3 WARC file index	5 WARC file index
Single URL get	0.001 sec.	0.002 sec.	0.003 sec.
HTML Query	2.784 sec.	10.525 sec.	17.067 sec.
JPG Query	2.62 sec.	10.311 sec.	17.151 sec.
UK HTML Query	2.91 sec.	10.433 sec.	17.130 sec.
RU PNG Query	2.55 sec	10.312 sec.	17.007 sec



Conclusion

- WARC-KIT a WARC toolkit created in JavaScript
- Provides a standalone WARC parsing tool in JavaScript that can create new WARC files, create CDX index files, and create Web graph datasets.
- Also, provides a Linear Hash Table database that provides document style storage.
- Finally, WARC-KIT's main function is to create custom indices upon a WARC collection for querying.

Future Work

- Web crawler in Node.js.
- Create better WARC data.
- Improve Hash table performance by further optimizing bucket splits.
- Improve Web graph dataset creation by filtering out Content delivery networks (CDNs).
- Improve WARC-KIT's GraphQL schema.

Thank You!

- Happy Holidays!

References

- 1] Panchal Akshar. Overlapping Community Detection in Social Networks. San Jose State University, 2021.
- [2] CDX and DAT Legend. url: https://archive.org/web/researcher/cdx legend.php (visited on11/10/2021).
- [3] Common Crawl Data. url: https://commoncrawl.org/the-data/get-started/ (visited on11/14/2021).
- [4] Dynamic hashing technique of Berkeley DB. url: https://titanwolf.org/Network/Articles/Article?AID=9823fa36-325a-40bc-99bc-8f6e0173be50 (visited on 11/14/2021).
- [5] GraphQL. url: https://graphql.org/ (visited on 11/14/2021).
- [6] Adi Robertson. Link rot in 2012: keeping track of how web addresses go dead. May 15, 2012. url: https://www.theverge.com/2012/5/15/3021913/chesapeake-digital-preservation-group-link-rot-report (visited on 11/10/2021).
- [7] RustLinearHashTableimplmentation.url:https://github.com/samrat/rust-linhash(visitedon 11/17/2021).
- [8] Stanford Web Archiving Tutorials and Resources. url: https://library.stanford.edu/projects/
- web-archiving/research-resources/tutorials-and-examples (visited on 11/15/2021).
- [9] WARC-KIT Code. url: https://www.cs.sjsu.edu/faculty/pollett/masters/Semesters/
- Spring21/david/WARC_KIT_Code.html.
- [10] WARC,WebARChivefileformat.url:https://iipc.github.io/warc-specifications/specifications/
- warc-format/warc-1.0/#warc-record-types (visited on 11/10/2021).
- [12] Yioop: Open Source Search Engine Software. url: https://www.seekquarry.com/ (visited on11/10/2021).