

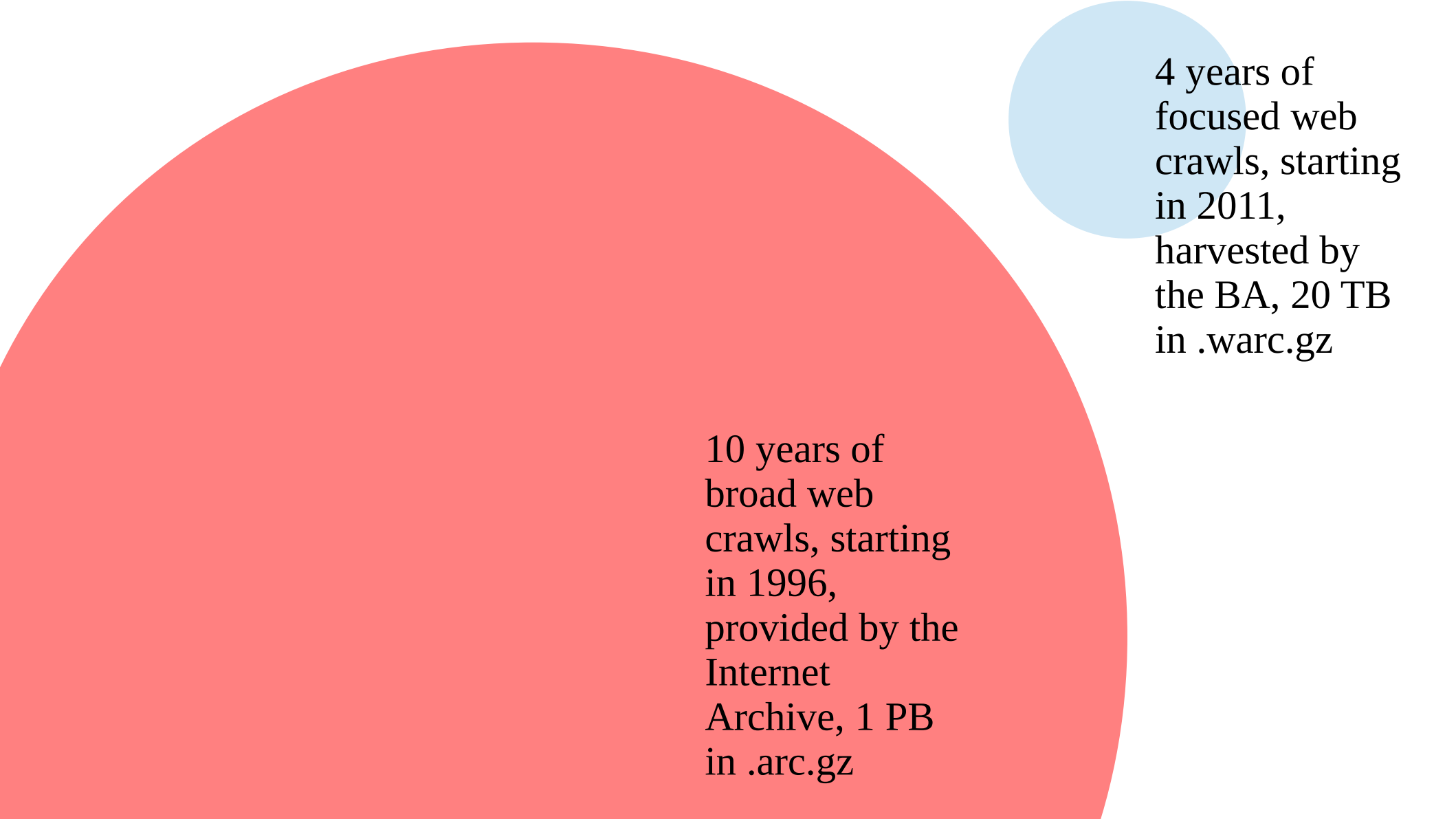


**BIBLIOTHECA ALEXANDRINA**

**مكتبة الإسكندرية**



# The BA web archive



4 years of  
focused web  
crawls, starting  
in 2011,  
harvested by  
the BA, 20 TB  
in .warc.gz

10 years of  
broad web  
crawls, starting  
in 1996,  
provided by the  
Internet  
Archive, 1 PB  
in .arc.gz

# The problem we are working to solve



Over time and across distance, the process of crawling the web produces collections with duplicates, i.e., records whose content is identical to an original copy

# How much duplication is in the collection?

- The BA sampled one-tenth of the legacy collection and calculated the rate of duplication to be approx. 14%
- Other IIPC members have reported even more significant rates in meetings
- Considering how data requires at least double the storage space for redundancy, working to eliminate duplication is well worth the effort

# How to identify a duplicate?

- Use a hash function
- Well-known algorithms: MD5, SHA-1, SHA-2
- MD5 is prone to *collisions*, there is a theoretical attack for SHA-1, SHA-2 is so far trustworthy
- There is a price to be paid: computation time, MD5 is the fastest
- Modern file systems implement deduplication, e.g., Btrfs, ZFS; ZFS uses 256-bit SHA-2 and runs *collision resolution*

# How to become unique

- While crawling
- Heritrix modules to stop processing of duplicate resources (Sigurðsson 2006)
- “After the fact”
- Why?
  - Legacy data has already been crawled a long time ago
  - No extra load on crawler instance
  - Run collision resolution

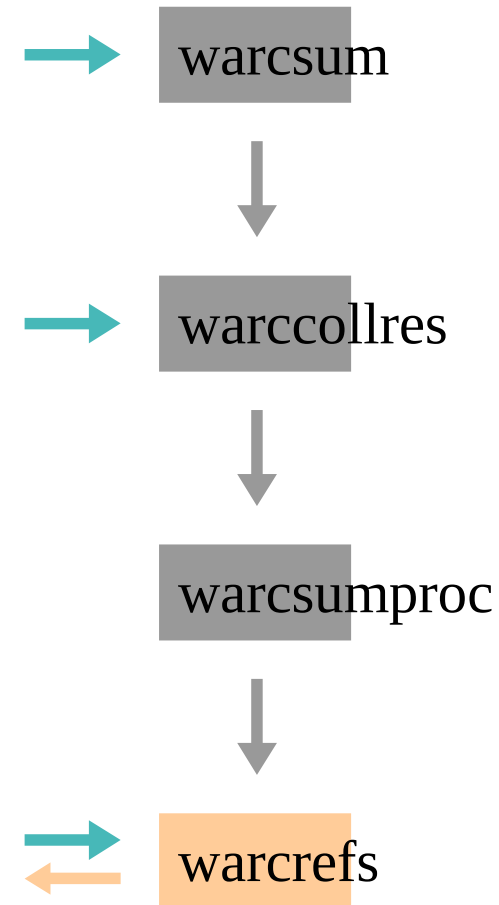
# A duplicate record in WARC

- See “Proposal for Standardizing the Recording of Duplicates in WARC Files,” IIPC Harvesting Working Group (2013)
- For each duplicate record, write a *revisit* record instead with *references* to the original copy:
  - WARC-Refers-To-Target-URI
  - WARC-Refers-To-Date



# WARCrefs for deduplicating web archives

- Post-crawl deduplication tools, 2 packages
- **WARCsum** does hash manifest generation and collision resolution
- **WARCrefs** provides the actual deduplicator
- [github.com/arcalex/](https://github.com/arcalex/)



hash manifest

warcsum

file offset length uri date [hash](#)

Choose which hash algorithm from OpenSSL to use: MD5, SHA-1, SHA-256, SHA-512

Fields in [blue](#) are sort keys

# Example hash manifest

3dCjphOZlL.warc.gz	<i>file</i>
20276	<i>offset</i>
4889	<i>length</i>
<a href="http://www.akhbarak.net/articles/10473209-...">http://www.akhbarak.net/articles/10473209-...</a>	<i>uri</i>
2012-12-12T07:59:14Z	<i>date</i>
sha1:0d92938a1a322622a5e95bffdfa023f1830da69a	<i>hash</i>

extended  
hash manifest

---

warccollres

file offset length uri date [hash](#) [ext](#)

Fetch records to  
compare via HTTP  
(just like the  
OpenWayback)

Fields in [blue](#) are sort keys

post-processed  
hash manifest

---

warcsumproc

file offset length uri date hash ext copy ref\_uri ref\_date

Fields in blue are sort keys

---

warcrefs

Deduplicated  
WARC file  
with revisit  
records



# What Next?

- Code review, minor feature implementation
- Testing, finding scenarios that would result in damage to data
- Executing the deduplication process on real data



**BIBLIOTHECA ALEXANDRINA**

**مكتبة الإسكندرية**