

JUBAKO

A SMALL INTRO

A BIT OF CONTEXT

WHO AM I ?

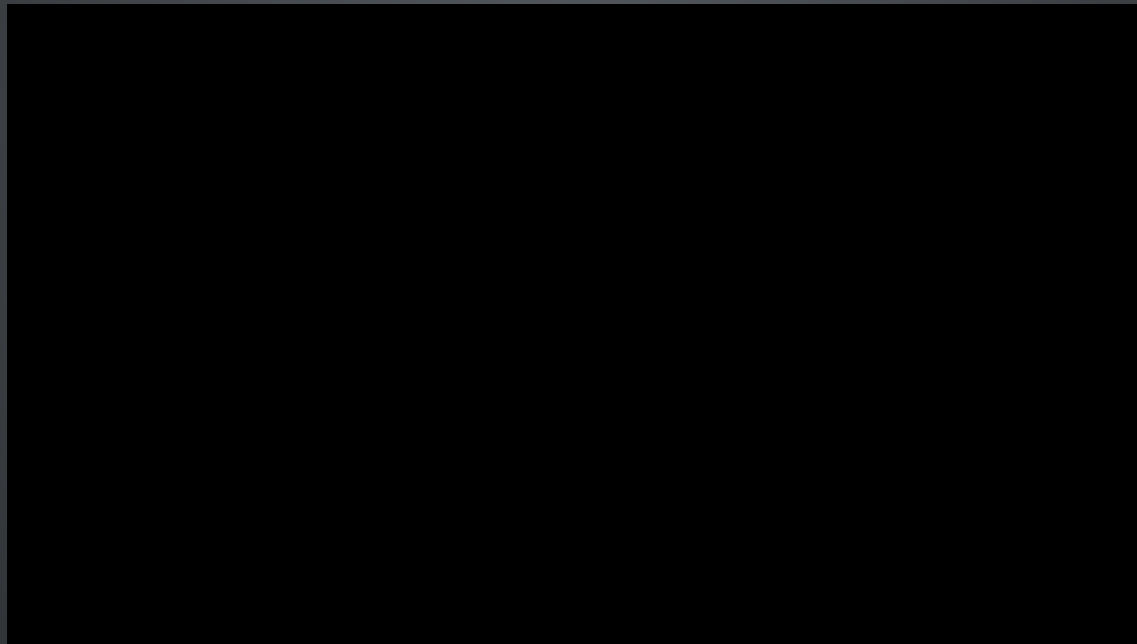
- Matthieu Gautier
- Freelance developer
- mgautier@kymeria.fr
- Lead developer of libzim (@kiwix)



KIWIX ?

www.kiwix.org

Provide knowledge where internet is not there



HOW TO DISTRIBUTE STATIC WEBSITE ?

HOW TO DISTRIBUTE STATIC WEBSITE ?

wikipedia_en_all_maxi_2022_05.zim

- 95GB
- 6 492 232 articles
- 6 693 406 media files

ZIM FORMAT

- An archive of web content(html/js/css/img/...).
- Partially compressed
- Random access without initial decompression

It works well and is pretty efficient.

BUT...

- Few flaws in the design
- Tied to web content (mimetype)
- Tied to kiwix (article's title, fulltext search index)
- No other metadata

Could we reuse that and do better ?

JŪBAKO (重箱)

The Japanese name for bento's boxes



Compartmented boxes you can compose the way you want depending of the food of the day.

JUBAKO

A new format independent of Kiwix.

The good ideas of zim format but generic.

Jubako is a meta-container:

It tells you how to store things,
not what to store and how to organize them.

A reference library written in Rust.

FEATURES

- Read only
- Selective compression
- No initial decompression needed
- Random access
- Configurable
- Extension system
- Embeddable in other file
- Composable
- Checksum

FEATURES (TO BE DONE)

(OR NOT)

- Signature/Encryption
- Direct access to uncompressed content
- Deduplication
- Modification
- Diff/Patch
- Overlay

INTERNAL STRUCTURE

A REALLY FAST TOUR

PACKS

- Manifest pack
- Content pack
- Directory pack

Can be stored individually as file or put in one file.

MANIFEST PACK

The main pack.

This IS the jubako container.

List all other packs part of a container.

CONTENT PACK

Store the raw content,
compressed or not,
without metadata.

DIRECTORY PACK

Store the "entries",
their metadata
and point to raw contents in contentPack.

The configurable part of Jubako.

ENTRY SCHEMA

You define your own schema:

- A series of properties (and their types)
- Content is just a property
- Can contain variants (as enum/union)

You can have different kinds of entry (schema) in one directory pack.

WHICH USE CASES ?

FILE ARCHIVE

ARX

File	SymLink	Directory
Base name (char array)		
parent id (idx)		
content address	link target (char array)	first entry (idx)
		nb children (uint)

LINUX SOURCES

	File system	Tar	Arx	Ratio
Size	1326 MB	129 MB	140 MB	0.92
Creation		1m40s	1m20s	1.25
Extration		1s30	1s50	0.80
Listing		900ms	50ms	18

LINUX SOURCES

	File system	Tar	Arx	Ratio
Size	1326 MB	129 MB	140 MB	0.92
Creation		1m40s	1m20s	1.25
Extraction		1s30	1s50	0.80
Listing		900ms	50ms	18
Dumping entries		5h20m	1m50s	175
time/entry		700ms	4ms	175

LINUX SOURCES

	File system	Tar	Arx	Ratio
Size	1326 MB	129 MB	140 MB	0.92
Creation		1m40s	1m20s	1.25
Extraction		1s30	1s50	0.80
Listing		900ms	50ms	18
Dumping entries		5h20m	1m50s	175
time/entry		700ms	4ms	175
(mount) Diff	800ms	~10h	4s5	>5000

LINUX SOURCES

	File system	Tar	Arx	Ratio
Size	1326 MB	129 MB	140 MB	0.92
Creation		1m40s	1m20s	1.25
Extraction		1s30	1s50	0.80
Listing		900ms	50ms	18
Dumping entries		5h20m	1m50s	175
time/entry		700ms	4ms	175
(mount) Diff	800ms	~10h	4s5	>5000
Compilation	32m		48m	0.66

STATIC WEB SITE

JIM

Content	Redirection
Path (char array)	
content address	link target (char array)
mimetype	

Entries are stored as a plain list.

Jim binary integrate a http server, looking for entry based on the request path.

SOFTWARE DISTRIBUTION

REPLACEMENT OF RPM/DEB

No need to extract the package on your FS.
Mount it on demand.

* - devel or debug - info could be put in a specific contentPack of the same archive.

OCI CONTAINER

OCI container are based on tar.

We could use arx and mount them.

Each layer could be stored in a content pack.

FILE FORMAT

`.odt`/`.doc`/`epub` are zip files containing
xml/image/videos.

`.xcf`(gimp) files contain several graphical layers.

A web page.

Presentations are composed of several contents (text,
img, video)

Most advanced formats are containers for other data.

WEB SITES

Jubako is written in rust, so it could run in wasm.

Load a Jubako archive once and read it in your browser.

BACKUPS

Jubako is almost incremental by design.

EMBEDDING RESOURCES

Jubako can be embedded in executable programs.

THIS PRESENTATION

https://public.kymeria.fr/jbk_love_fossdem

A ARX ARCHIVE

- `arx list jbk_love_fosdem`
- `arx extract jbk_love_fosdem A_DIR`
- `arx mount jbk_love_fosdem MOUNT_POINT`

A JIM ARCHIVE

- `jim serve jbk_love_fosdem localhost:8000`

A PROGRAM

- `./jbk_love_fosdem mount MOUNT_POINT`
- `./jbk_love_fosdem extract DIRECTORY`
- `./jbk_love_fosdem serve localhost:8000`

CONCLUSION

CONCLUSION

A NEW WAY OF THINKING

Use the archive directly and don't extract it.
Reinvent ~~tools~~ wheels using Jubako !

CONCLUSION

A NEW WAY OF THINKING

Use the archive directly and don't extract it.
Reinvent ~~tools~~ wheels using Jubako !

GENERIC

A common code base which adapt for different usages.

CONCLUSION

A NEW WAY OF THINKING

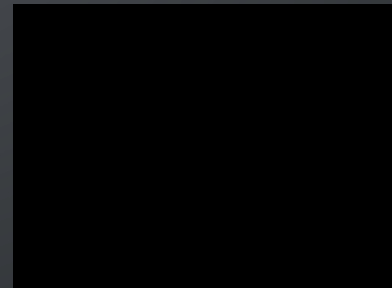
Use the archive directly and don't extract it.
Reinvent ~~tools~~ wheels using Jubako !

GENERIC

A common code base which adapt for different usages.

NEW

Not production ready.



THANKS

- <https://framagit.org/jubako/> (MIT + GPLv3)
- https://public.kymeria.fr/jbk_loves_fosdem.pdf
- pdf (local)
- mgautier@kymeria.fr