JUBAKO
A SMALL INTRO
A BIT OF CONTEXT
WHO AM I?

- Matthieu Gautier
- Freelance developer
- mgautier@kymeria.fr
- Lead developper 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

Compartimented boxes you can compose the way you want depending on 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**

<table>
<thead>
<tr>
<th>File</th>
<th>SymLink</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base name (char array)</td>
<td>parent id (idx)</td>
<td></td>
</tr>
<tr>
<td>content address</td>
<td>link target (char array)</td>
<td>first entry (idx)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>nb children (uint)</td>
</tr>
</tbody>
</table>
## LINUX SOURCES

<table>
<thead>
<tr>
<th></th>
<th>File system</th>
<th>Tar</th>
<th>Arx</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td>1326 MB</td>
<td>129 MB</td>
<td>140 MB</td>
<td>0.92</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>1m40s</td>
<td>1m20s</td>
<td>1.25</td>
<td></td>
</tr>
<tr>
<td><strong>Extration</strong></td>
<td>1s30</td>
<td>1s50</td>
<td>0.80</td>
<td></td>
</tr>
</tbody>
</table>
## Linux Sources

<table>
<thead>
<tr>
<th></th>
<th>File System</th>
<th>Tar</th>
<th>Arx</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>1326 MB</td>
<td>129 MB</td>
<td>140 MB</td>
<td>0.92</td>
</tr>
<tr>
<td>Creation</td>
<td>1m40s</td>
<td>1m20s</td>
<td></td>
<td>1.25</td>
</tr>
<tr>
<td>Extration</td>
<td>1s30</td>
<td>1s50</td>
<td></td>
<td>0.80</td>
</tr>
<tr>
<td>Listing</td>
<td>900ms</td>
<td>50ms</td>
<td></td>
<td>18</td>
</tr>
</tbody>
</table>
### LINUX SOURCES

<table>
<thead>
<tr>
<th></th>
<th>File system</th>
<th>Tar</th>
<th>Arx</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>1326 MB</td>
<td>129 MB</td>
<td>140 MB</td>
<td>0.92</td>
</tr>
<tr>
<td>Creation</td>
<td>1m40s</td>
<td>1m20s</td>
<td></td>
<td>1.25</td>
</tr>
<tr>
<td>Extration</td>
<td>1s30</td>
<td>1s50</td>
<td></td>
<td>0.80</td>
</tr>
<tr>
<td>Listing</td>
<td>900ms</td>
<td>50ms</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>Dumping entries</td>
<td>5h20m</td>
<td>1m50s</td>
<td></td>
<td>175</td>
</tr>
<tr>
<td>time/entry</td>
<td>700ms</td>
<td>4ms</td>
<td></td>
<td>175</td>
</tr>
<tr>
<td></td>
<td>File system</td>
<td>Tar</td>
<td>Arx</td>
<td>Ratio</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>Size</td>
<td>1326 MB</td>
<td>129 MB</td>
<td>140 MB</td>
<td>0.92</td>
</tr>
<tr>
<td>Creation</td>
<td>1m40s</td>
<td>1m20s</td>
<td></td>
<td>1.25</td>
</tr>
<tr>
<td>Extration</td>
<td>1s30</td>
<td>1s50</td>
<td></td>
<td>0.80</td>
</tr>
<tr>
<td>Listing</td>
<td>900ms</td>
<td>50ms</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>Dumping entries</td>
<td>5h20m</td>
<td>1m50s</td>
<td></td>
<td>175</td>
</tr>
<tr>
<td>time/entry</td>
<td>700ms</td>
<td>4ms</td>
<td></td>
<td>175</td>
</tr>
<tr>
<td>(mount) Diff</td>
<td>800ms</td>
<td>~10h</td>
<td>4s5</td>
<td>&gt;5000</td>
</tr>
</tbody>
</table>
## LINUX SOURCES

<table>
<thead>
<tr>
<th></th>
<th>File system</th>
<th>Tar</th>
<th>Arx</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td>1326 MB</td>
<td>129 MB</td>
<td>140 MB</td>
<td>0.92</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>1m40s</td>
<td>1m20s</td>
<td></td>
<td>1.25</td>
</tr>
<tr>
<td><strong>Extration</strong></td>
<td>1s30</td>
<td>1s50</td>
<td></td>
<td>0.80</td>
</tr>
<tr>
<td><strong>Listing</strong></td>
<td>900ms</td>
<td>50ms</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td><strong>Dumping entries</strong></td>
<td>5h20m</td>
<td>1m50s</td>
<td></td>
<td>175</td>
</tr>
<tr>
<td><strong>time/entry</strong></td>
<td>700ms</td>
<td>4ms</td>
<td></td>
<td>175</td>
</tr>
<tr>
<td><strong>(mount) Diff</strong></td>
<td>800ms</td>
<td>~10h</td>
<td>4s5</td>
<td>&gt;5000</td>
</tr>
<tr>
<td><strong>Compilation</strong></td>
<td>32m</td>
<td>48m</td>
<td></td>
<td>0.66</td>
</tr>
</tbody>
</table>
# Static Web Site

**Jim**

Entries are stored as a plain list.

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

<table>
<thead>
<tr>
<th>Content</th>
<th>Redirection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path (char array)</td>
<td>link target (char array)</td>
</tr>
<tr>
<td>content address</td>
<td></td>
</tr>
<tr>
<td>mimetype</td>
<td></td>
</tr>
</tbody>
</table>
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.

.xfc (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.
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.
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

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)
- pdf (local)
- mgautier@kymeria.fr