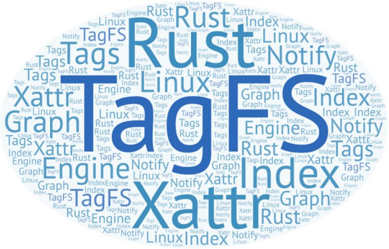
TagFS - File Labeling System with Rust

Steven Liatti

Bachelor project - Prof. Florent Glück - Hepia ITI 3rd year

September 4, 2018



hepia

Haute école du paysage, d'ingénierie
et d'architecture de Genève

Hes·so//GENÈVE

Haute Ecole Spécialisée
de Suisse occidentale

Plan

1. Introduction
2. Existing solutions
3. Architecture
4. Technologies
5. Achievement
6. Discussion
7. Conclusion

Issues

- Huge number of files.

Issues

- Huge number of files.
- Difficulty finding files.

Issues

- Huge number of files.
- Difficulty finding files.
- Multiple logical locations for a single file.

Issues

- Huge number of files.
- Difficulty finding files.
- Multiple logical locations for a single file.

System for tagging files and directories with the possibility of searching by tags.

Specifications

- List existing applications for tagging files.

Specifications

- List existing applications for tagging files.
- Investigate a way to store tags with files: Extended File Attributes (XATTR).

Specifications

- List existing applications for tagging files.
- Investigate a way to store tags with files: Extended File Attributes (XATTR).
- Analyze ways to index and monitor a tree of files.

Specifications

- List existing applications for tagging files.
- Investigate a way to store tags with files: Extended File Attributes (XATTR).
- Analyze ways to index and monitor a tree of files.
- Design and implement the system (open source and on Linux) and measure its performance.

Specifications

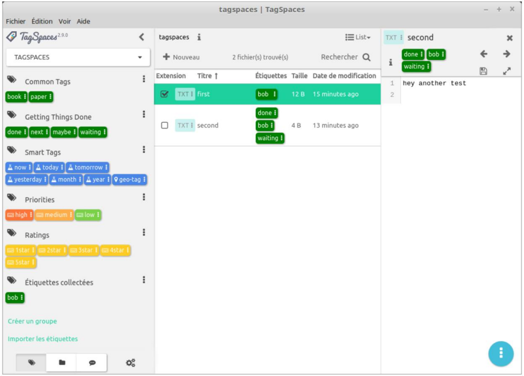
- List existing applications for tagging files.
- Investigate a way to store tags with files: Extended File Attributes (XATTR).
- Analyze ways to index and monitor a tree of files.
- Design and implement the system (open source and on Linux) and measure its performance. Study and learn the Rust language.

TMSU, Tagsistant et TagSpaces

- Tag management.
- List of files related to tags.
- CLI or GUI.

TMSU, Tagsistant et TagSpaces

- Tag management.
- List of files related to tags.
- CLI or GUI.



Extension	Titre	Étiquettes	Taille	Date de modification
☑ TXT	first	bob	12 B	15 minutes ago
□ TXT	second	done bob waiting	4 B	13 minutes ago

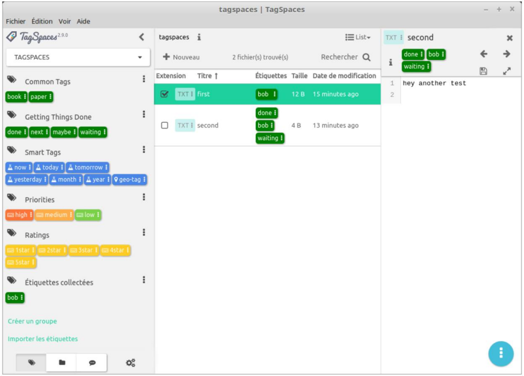
```
hey another test
```

Figure – Using TagSpaces

TMSU, Tagsistant et TagSpaces

- Tag management.
- List of files related to tags.
- CLI or GUI.

Good points	Negative points
Simple	DB dependency outer
Fast and efficient hunt	Modification and access only through the app
Open source	



Extension	Titre ↑	Étiquettes	Taille	Date de modification
☑ TXT	first	bob	12 B	15 minutes ago
□ TXT	second	done bob waiting	4 B	13 minutes ago

```
hey another test
```

Figure – Using TagSpaces

macOS

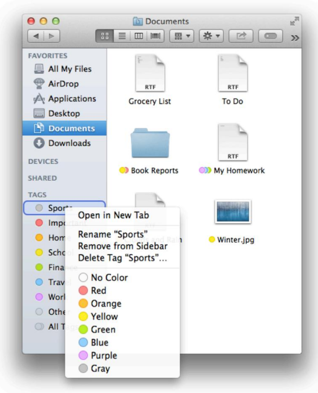


Figure - Managing a tag in the Finder - Apple

macOS

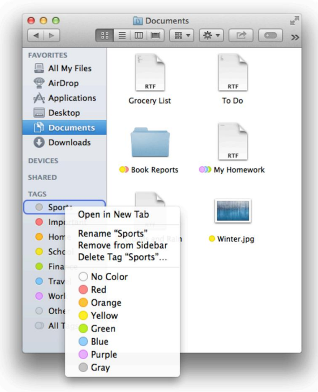


Figure - Managing a tag in the Finder - Apple

Good points	Negative points
Tag system integrated into file explorer	Owner code
Store tags in XATTR	Only for macOS
efficient	

Tag management

Tag management

- Storage of tags with files (independence from a database).

Tag management

- Storage of tags with files (independence from a database).
- Dedicated tag management tool (easy to use).

Indexing files and tags

Indexing files and tags

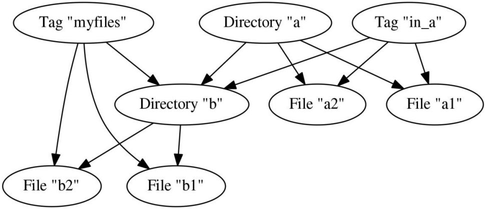


Figure – Graph of tags, files and directories

Indexing files and tags

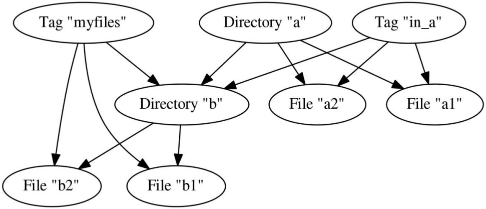


Figure – Graph of tags, files and directories

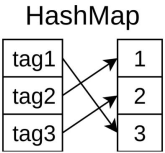


Figure – HashMap associating the name of the tag to its node

File system monitoring

Update of the graph during the following events:

File system monitoring

Update of the graph during the following events:

- Change on tags.

File system monitoring

Update of the graph during the following events:

- Change on tags.
- Creation of files/directories.

File system monitoring

Update of the graph during the following events:

- Change on tags.
- Creation of files/directories.
- Delete files/directories.

File system monitoring

Update of the graph during the following events:

- Change on tags.
- Creation of files/directories.
- Delete files/directories.
- Moving/renaming files/directories.

Tags and files queries

Tags and files queries

- List the files and directories associated with tags => queries in the form of logical expressions (logical operators).

Tags and files queries

- List the files and directories associated with tags => queries in the form of logical expressions (logical operators).
- List existing tags.

Tags and files queries

- List the files and directories associated with tags => queries in the form of logical expressions (logical operators).
- List existing tags.
- Rename a tag.

Rust

- Modern system language, efficient, reliable (more secure than Ada), compiled, and strongly typed.

Rust

- Modern system language, efficient, reliable (more secure than Ada), compiled, and strongly typed.
- Available on Linux, Windows and macOS.

Rust

- Modern system language, efficient, reliable (more secure than Ada), compiled, and strongly typed.
- Available on Linux, Windows and macOS.
- Cargo: Rust's built-in build and runtime tool and package manager.

Rust

- Modern system language, efficient, reliable (more secure than Ada), compiled, and strongly typed.
- Available on Linux, Windows and macOS.
- Cargo: Rust's built-in build and runtime tool and package manager.
- Structures, collections, genericity, immutability, enumerations and pattern matching.

Rust

- Modern system language, efficient, reliable (more secure than Ada), compiled, and strongly typed.
- Available on Linux, Windows and macOS.
- Cargo: Rust's built-in build and runtime tool and package manager.

- Structures, collections, genericity, immutability, enumerations and pattern matching.
- Error handling and unit testing.

Rust

- Modern system language, efficient, reliable (more secure than Ada), compiled, and strongly typed.
- Available on Linux, Windows and macOS.
- Cargo: Rust's built-in build and runtime tool and package manager.

- Structures, collections, genericity, immutability, enumerations and pattern matching.
- Error handling and unit testing.
- Ownership, Borrowing (references).

Extended Attributes (XATTR)

Extended Attributes (XATTR)

- Metadata as name:value pairs.

Extended Attributes (XATTR)

- Metadata as name:value pairs.
- Name = character string, value = character string or binary data.

Extended Attributes (XATTR)

- Metadata as name:value pairs.
- Name = character string, value = character string or binary data.
- Existent sous ext2-3-4, XFS, Btrfs, UFS1-2, NTFS, HFS+, ZFS.

Extended Attributes (XATTR)

- Metadata as name:value pairs.
- Name = character string, value = character string or binary data.

- Existent sous ext2-3-4, XFS, Btrfs, UFS1-2, NTFS, HFS+, ZFS.
- CLI tools to easily manipulate them.

Inotify

Inotify

- File system event notification API.

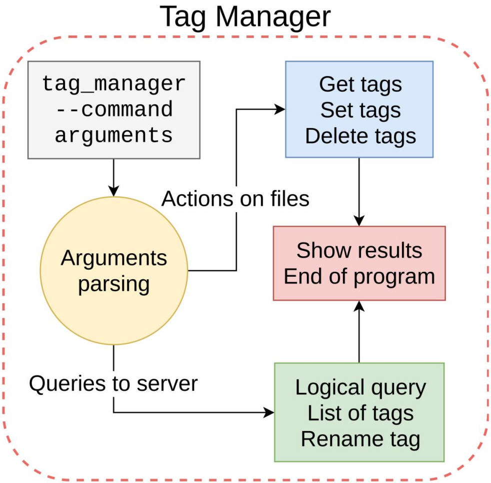
Inotify

- File system event notification API.
- Three system calls: initialization, adding watch on a given file path, and removing that watch.

Inotify

- File system event notification API.
- Three system calls: initialization, adding watch on a given file path, and removing that watch.
- Reading an event with read().

Tag Manager



Tag Manager

tag_manager --command arguments

Arguments parsing

Actions on files

Get tags
Set tags
Delete tags

Show results
End of program

Queries to server

Logical query
List of tags
Rename tag

Figure – How Tag Manager works

Tag Engine

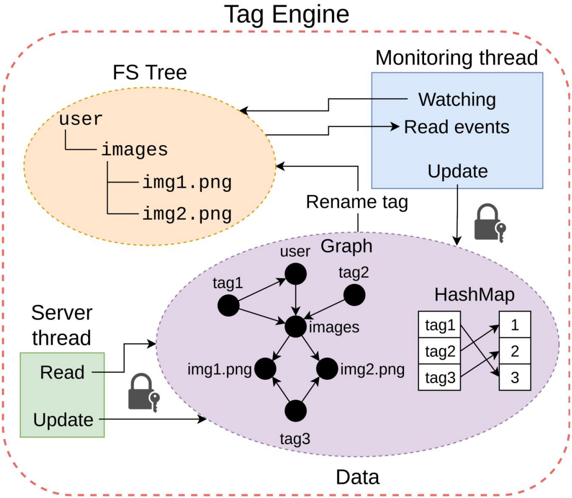


Figure – How Tag Engine works

TagFS

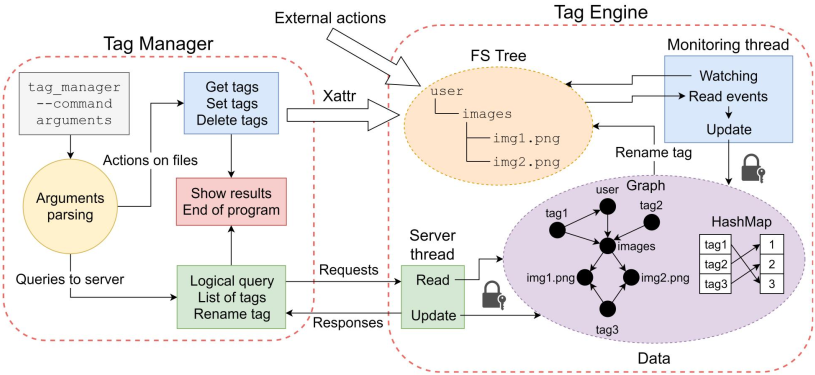
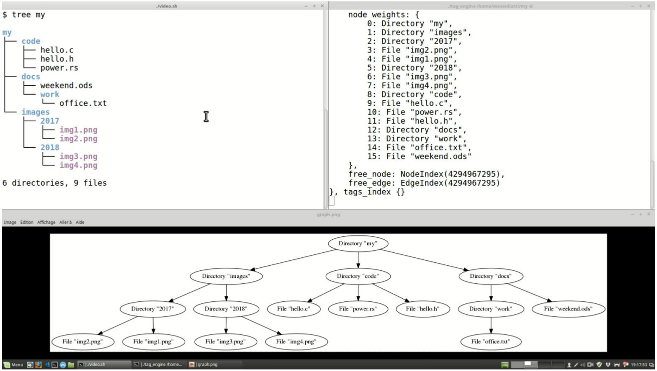


Figure – TagFS Global System

Demo

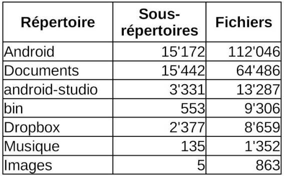


```
$ tree my

my
├── code
│   ├── hello.c
│   ├── hello.h
│   └── power.rs
├── docs
│   ├── weekend.ods
│   └── work
│       └── office.txt
└── images
    ├── 2017
    │   ├── img1.png
    │   └── img2.png
    └── 2018
        ├── img3.png
        └── img4.png

6 directories, 9 files
```

```
    node weights: {
        0: Directory "my",
        1: Directory "images",
        2: Directory "2017",
        3: File "img2.png",
        4: File "img1.png",
        5: Directory "2018",
        6: File "img3.png",
        7: File "img4.png",
        8: Directory "code",
        9: File "hello.c",
        10: File "power.rs",
        11: File "hello.h",
        12: Directory "docs",
        13: Directory "work",
        14: File "office.txt",
        15: File "weekend.ods"
    },
    free_node: NodeIndex(4294967295),
    free_edge: EdgeIndex(4294967295)
}, tags_index {}
```

Video

Performance Metrics



Répertoire	Sous-répertoires	Fichiers
Android	15'172	112'046
Documents	15'442	64'486
android-studio	3'331	13'287
bin	553	9'306
Dropbox	2'377	8'659
Musique	135	1'352
Images	5	863

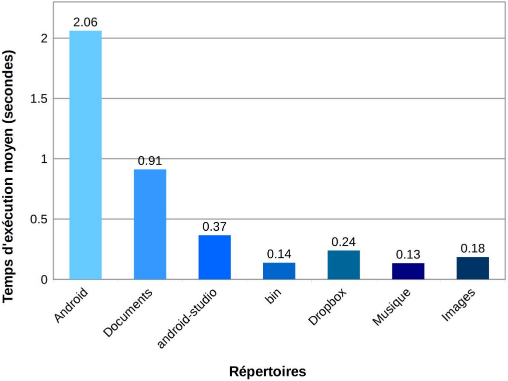


Figure – Tag Engine runtime based on directories

Improvements

Improvements

- GUI: desktop environment or web application.

Improvements

- GUI: desktop environment or web application.
- Daemon pour Tag Engine.

Improvements

- GUI: desktop environment or web application.
- Daemon pour Tag Engine.
- Added new watch directories (partial).

Improvements

- GUI: desktop environment or web application.
- Daemon pour Tag Engine.
- Added new watch directories (partial).
- Management of removable devices (inotify limitation).

Improvements

- GUI: desktop environment or web application.
- Daemon pour Tag Engine.
- Added new watch directories (partial).
- Management of removable devices (inotify limitation).
- Cache of the last logical requests sent to the server.

Improvements

- GUI: desktop environment or web application.
- Daemon pour Tag Engine.
- Added new watch directories (partial).
- Management of removable devices (inotify limitation).
- Cache of the last logical requests sent to the server.
- Add logical operators (NOT).

Bilan personnel

Bilan personnel

- Design and implementation of a tag management engine.

Bilan personnel

- Design and implementation of a tag management engine.
- Study of the Rust language.

Bilan personnel

- Design and implementation of a tag management engine.
- Study of the Rust language.
- Personal progress: new technologies and best practices.

Thanks

- Florence luck
- Orestis Malaspina
- Joel Cavat

Thanks

- Florence luck
- Orestis Malaspina
- Joel Cavat

Thank you for your attention ! Issues ?