Report 6

Hivi-Repo specification

Le Nhu Chu Hiep

August 19, 2020

1. Introduction

This report is a part of the hivilake project. Firstly, the "hivi-repo" is a binary file storage model orienting metadata. It provides a consistent self-describe folder to hold and keep track multiple binary file at once. This model was invented to provide the linage, audit and discoverable properties for the binary file management system.

2. Objective

This spcification proposes a model for non-modify self-describe data holder supporting governance process. To do that, this model applies several tracking mechanism:

- History tracking
- File describe tracking
- Folder consistency tracking

These tracking mechanism supports the data discovery process as well as prevents the system crash caused during update task.

3. Methodology

Hivilake system aims to store binary file. And the binary has hard manage and discovery content, instead, metadata keep more important role to keep track and discovery data. Therefore, the concern of the binary file management system will be the file metadata.

Since A good binary file management should provide the useful trusted metadata of file that it manages. The "hivi repo" allow it through the key concept: "centralized metadata file". By splitting metadata out of their file and keep them in a centralized file, the model help the binary manage process becomming super easy.

3.1 Repo Structure

To implement its key concept, the "hivi repo" defined itself as a folder (repo) which holds all file sharing same metadata attributes. Since each file in repo uses same metadata schema, the centralized metadata file can be constructed as a tabular file. And repo shape will like:

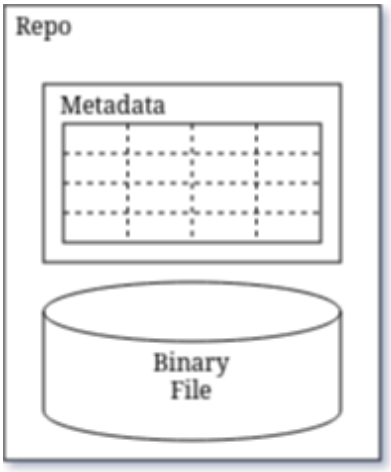

Figure 1: Repo Shape

The strength of this design groups the multiple file metadata in to a tabular file which can leverage the sql power to control. And since it split the binary content out of its describe, the provenance process can skip binary data loading and directly work with the meta content. It allow speeding task a lots. If the binary content needed, its linking path can be extracted from metadata file.

3.2 Metadata Schema

The model expected to enable the file describe and history tracking ability. So the file metadata will keep information to support them. Json object below describe detail metadata schema:

```
{
    # history tracking
    user        : user_id,
    time        : time_upload (yyyy-mm-dd T hh:mm:ss),

    # file describe
    name        : file_name,
    type        : type_of_file (dir/file),
    format      : file_format,
    label       : list_label_of_file (increase integer seperated by space),
    description : file_description,
    path        : link_to_real_file,

    # audit
    a_type      : boolean,
    a_format    : boolean,
    size        : file_size,
    status      : file_status (good/corrupt/non-exist),
    meta_null   : null_percent_of_meta_fields,

    # user field
    u_<field_1> : field_1,
    u_<field_2> : field_2,
    u_<field_3> : field_3,
    ...
    u_<field_n> : field_n
}
```

This schema provide enough field for “hitory tracking” and “file describe”. Moreover, it allows user can define their

own fields following a convention that every customize field must initialize with “u_” string as the recognizable value.

3.3 Folder Consistency

Although the metadata provide keep track mechanism for all file in its folder. This file itself can be crashed specialy in the concurrent environment. Therefore, the model advise the implementation should include a metadata backup strategy as well as lock mechanism.

Another problem is folder healthy. The “hivi repo” model not only tracks file ingest, it also provides a folder monitor method through a tracking file as:

```
# repo_tracking.json
{
    name        : repo_name,
    description : repo_describe,
    number      : repo_file_number,
    size        : repo_size,
    create_time : repo_create_time,
    update_time : repo_update_time,
    audit_time  : repo_final_audit_time
}
```

3.4 Binary Content Convention

Since the real binary file will be stored as a normal file in filesystem, the model provides the way naming each file. The “hivi repo” follows convention mentioned in section “3.2.1” path “SM Exception” of report 5, the binary file will be named by combining 3 special key with:

```
file_user_id + time_create + file_name + ".bin"(seperate by "_" charater)
================================================================
Example:
0_2020-03-13T00:05:02_binary-file.bin
```

This convention as mentioning in report 5 help reducing the probability of file name duplicate during ingestion time on concurrent environment and also providing a basic recovery information from file name.