Peter Deng
FALL 2012
Undergraduate Research

Media Khan Report

Project Goal

The initial overall goal of this research project is to be able to create a filesystem that will be able to automatically characterize, manage, and sort out files based on various meaningful metadata as well as the current situation. In the scope of this semester, the area of focus was delimited to having the file system handle only music and pictures files and exploring metadata organization and methods to allow some headroom for user customization, such as creating a custom playlist using simple set operations. The first step going into the project regarded taking a look at the capabilities of the existing file system that was developed using performance metrics and determine any areas that might need to be patched up. The next step is to then focus on building up the filesystem and implementing/improving the ability to sort out and read files based on the metadata the individual files would be associated with. The end goal is to be able to demo the file system being able to select and organize music and picture files that would be appropriate for the user.

Media Khan Overall Details and Current State

In Media Khan's current state, it is able to read in jpg and mp3 (picture and music) files from a remote networked folder, utilize a database to hold metadata, sort out/organize the media files into respected attribute folder names, and allow users to generate semi-custom folders through the use of set operations (union and intersect).

The way Media Khan is managing the files is through metadata and pointers. Rather than storing the data of the actual files into folders and subfolders, we just have one copy of the said file sitting one media folder/partition. We could think of this as a soft link that points to a particular file. Metadata from music files is generated through the use of a third party binary/library called MP3INFO, and it will parse out information that we could use to help organize the music files. Metadata such as Artist, Genre, and Music Length can be extracted, provided that they are on the file, and the file system will then sort the files out, based on default configurations, and insert them into the database backend. To aid us in structuring out the file system, we used a "NoSQL" or "Key-Value Store" database called Redis. Voldemort, another "Key-Value Store" database, but due to problems I have encountered, it is not enabled in the current build. Another functionalities implemented in Media Khan are user folder/playlist customization and refresh. Specific implementation details will be provided later on, but in essence, the user is able to have some control over the file system such as creating a semi-custom playlist on-the-fly by conducting set operations on a group of files.

Performance Metrics

For the “Database Not Initialized” trial, I had the file system read in approximately 137 files (and totals to around 1GB) and pull these files through the network from 4sharded.com using WEBDAV. Redis started off clean with nothing recorded in the database. The result is a long wait for Khan to read in these files and extract metadata and put it in the database. The main cause for this is mainly due to network latency with 4shared.com.

For “Database Initialized” trial, an existing database dump of the session is preloaded. Media Khan, after seeing that the database is already initialized, skips the scan process and is ready to go with user operations.

For “LS Operations” trial, I conducted an “ls” on a folder to see how long it takes for Khan to get the contents of a folder. I did this on the folder that contained 130 music files.

The union operation trial involved merging/unioning together 20 items (1 folder with 6 items and 1 folder with 14 items). The intersect operation trial involved working with/intersecting 30 items (1 folder with 20 item and another folder with 10).

(NOTE) The measurements in the following table are measured in seconds.

Trials	Database Not Initialized (Cold)	Database Initialized (Warm)	LS Operations	Union Operation	Intersect Operation
1	770	0.24	5.17	0.024	0.038
2	810	0.22	5.31	0.022	0.033
3	902	0.16	5.65	0.023	0.030
4	825	0.20	5.25	0.022	0.034
5	854	0.18	5.05	0.021	0.032
AVG	832	0.20	5.26	0.022	0.033

Implementations Added, Rationale, and Results

Some functionality that I have added to this file system includes refreshing/updating the contents of the database to obtain an updated list of the list of files contained in the media folder, and adding set operations such as union and intersect to help group folders together. These added functionality addressed some of the issues I have encountered when using the file system.

One of the first problems I have encountered when using Media Khan is when I tried inserting new files into the media folder. When new files were added to the media folder, Media Khan was not able to immediately recognize the new file’s presence. In Media Khan’s previous state, it can read in files at the beginning but does not go back into rescan mode to refresh its contents later on. The command “mkdir refreshall” is essentially the code implementation of flushing the database and rescanning all the files in the media folder and activated by user command. As a result, we now have a way, through code implementation, to have Media Khan refresh it’s contents. However, this was initially meant to be a temporary fix, because the performance is analogous to doing a cold start all over again. What is preferred is to have the

file system be able to recognize that the media folder holding these files has its contents changed, find the file that was changed/inserted/deleted, and make the respected changes to the database. From the table above, doing a cold start is not an effective method as a user would not like to wait 10-15 min for the file system to finish loading its contents.

A functionality that I have implemented is the ability for the user to create a custom playlist. Previously, the user is able to create a folder within Media Khan and is able to copy a “file” within the Media Khan FS into that folder, but is tedious and unwieldy when doing it one by one. Since the file system is working on metadata and the metadata can be used to organize the files into respected folders based on the files attributes that describes the file, we could do set operations on these files and folders to combine similar attributed files together. Taking it up one more level, we can let the user create a semi-custom folder using these set operations. Basic unions and intersections can be done and the commands to do it are “mkdir FOLDERNAME.union.FOLDERNAME” and “mkdir FOLDERNAME.intersect.FOLDERNAME”. As for the results, the performance looks good as it does not take too much time.

Running the File System

To run and try out the file system, a media folder needs to be initially set up somewhere and should at least contain a number of music and/or picture files. To see a full list of steps please refer to “Media Khan Steps”; this document entails the steps that I have followed to make Media Khan as well as set up the necessary parts to have a working and running program.

Areas of Improvement

The time it takes to load in files from the network does vary in time quite a bit and is very long. Possible solutions would be to have say a thread that runs in the background that will slowly scan the media folder to check for any updates or to do lazy loading and load the files as they trickle in through the network.

Union and intersection set operations are only a few of the basic set operations, not to mention other complex set theory related stuff. Adding in more set operations would give the users more options, freedom, and customization of what to include and what not to include in a folder from the attributes given. This in turn would also give us more tools to use to reorganize the file system and/or work with in terms of metadata organization.

Areas to Explore

Since the file system is backed by a database of sorts, we could try and conduct some sort of query for a file given some attributes. When you run Media Khan and look through the directories, it is doing some sort of query, but it is doing it one attribute at a time. If we include set operations to help the user search for certain files, we can easily give the user a group of related files and could turn into a decent search engine for the file system.

So far, we have been using MP3INFO to help us generate metadata for the music files. We are currently not using a metadata generator for pictures yet and it would be awesome if we could also be able to characterize and generate metadata for movies and videos as well. If we can get a metadata generator for pictures and video, then Media Khan could be a full fledged

self-organizing media server/file system.

Although it is not covered in this scope of the project, Media Khan could read in files from multiple servers/locations. To be able to conduct some disconnect and reconnect with different devices and dynamically update it's contents will be quite a feat, but nonetheless interesting as it is essentially making the file system location aware.