Owen Voisinet
Northern Michigan University
ovoisine@nmu.edu
517-512-3038

Multi-Directory Data Integrity

A simple and lightweight alternative to RAID for data redundancy

---

Committee

Advisor: Dr. Randy R. Appleton

Dr. Michael R. Kowalczyk

Dr. Jeffrey Horn

The Goal

To create a piece of software that can be used to reduce data corruption and single points of failure from a storage system, specifically for those that do not require or can not afford a full RAID system. On a surface level, a Muti-Directory Data Integrity (MDDI) application is conceptually similar to a system employing RAID 1, which involves simply duplicating a hard drive and any changes to the stored information affect both drives. Unfortunately RAID has its limitations, being less flexible in implementation and often being prohibitively expensive for small operations. This project will bring about a free and open source piece of software that can still be useful even with limited hardware in order to enhance data integrity and reduce mistakes.

The Program

MDDI will have the ability to select or create a minimum of 3 directories and will be able to, at a regular intervals, scan through all directories file by file and bit by bit to check if the contents of all directories match. Whereas RAID 1 requires that these “directories” be separate, entire drives; MDDI will be able to work with any directory that the computer currently has access to. If all the user has available to them is a single hard drive, MDDI can still be used to prevent data rot. If the user selects directories on at least 3 separate hard drives, MDDI can be used to prevent data loss after physical hard drive failure, even failure due to outside means such as electrical surge if the directories are in physically separate locations and available over local or

network drives. In the event that information in the minority of directories does not match information in the majority of directories, the computer will automatically record the event and change the minority to match the majority. In the event that an entire file exists in a minority of directories and not in the majority, the event will be recorded, the file can be pulled and put into a backup directory to be looked over later, and the user can be notified. While this system cannot be used as a replacement for RAID, it can be used in niche applications to protect from data rot and physical drive failure or where a RAID implementation is not possible because of cost or flexibility.

Technologies Used

I will be using primairly C++ and necessary libraries to build both a Windows Graphical User Interface and backend. I will be putting this on Github in order to provide it as an open source project that can be added on to.

Primary Goals

- Graphical User Interface for controlling application
- Ability to choose 3 at least connected directories to check
- Start and Stop commands for the bit and file checking part of the program
- Ability and option to close the program to Windows tray
- Program stops bit and file checking when the minimum number of directories are no longer available
- Warning system for when drives fail or extra files are found
- Ability to forcefully sync all directories at the user's request
- Ability to change the check time and to find the minimum check time for a set of data
- Readme.txt file explaining what the program does, including a reminder that the program does not protect the user from themselves, it is intended to only protect the user from the bit rot and drive failure.

Secondary Goals

- Ability to add more than 3 directories
- The program should be able to operate on drives that are labeled as read only, that way drives can be protected from user interference while the program is running
- Email notification when problem occurs
- Ability to automatically halt program in case of failure to access a directory
- Ability to attempt to reconnect to network drives if initially unavailable