

Automations for Database Services Group

Abby Jurewicz – Michigan Technological University – SIST intern – Natalia Ratnikova, FNAL – FERMILAB-POSTER-23-169-STUDENT

Introduction

Database service group(DSG) is a part of Fermilab's IT Division that operates hundreds of databases which serve as the back-end for various applications that support the lab's mission and scientific experiments needs. This summer I was able to work for this group by developing automated tools in Python and Bash that help with movement of backups, enhancing the functionality of an existing tool and parsing of the database server log files.

Adding functionality to MyToolMan

There are two versions of MariaDB software on multiple host instances that showed up as one version. In order to support multiple versions, the two parts of code were modified:

- The Main loop
- The internal data structures

The tool now iterates through the configuration file and stores each database name and the corresponding version in a hash table. See figure 1. The new functionality was tested by the DSG team and deployed on the production server.

LogParser.py Script

Currently MongoDB monitoring connects to the server to get information. My task was to extract additional information from the log files. For example, for the operational security, DBAs want to know what hosts are connecting, how frequently, whether they succeed to authenticate.

Starting in MongoDB 4.4, all log messages are represented in structured JSON format, each containing a named component. I use messages in 'NETWORK' and 'ACCESS' components to extract the connection and authentication details. Results are presented in Fig.3. The connecting IP addresses are stored persistently using Python3 pickle module, so the next time the script runs, it can detect new IP addresses and alert the DBAs. Working with large log files, we found filtering with regular expressions works much faster than JSON objects. The processing time for the 4M lines (1GB) log file was reduced from 20 min to 5 sec. My next step is to write a tool to get a full catalog of MongoDB log messages, organized by component and severity. This script is going to be used on production host to help monitor connections

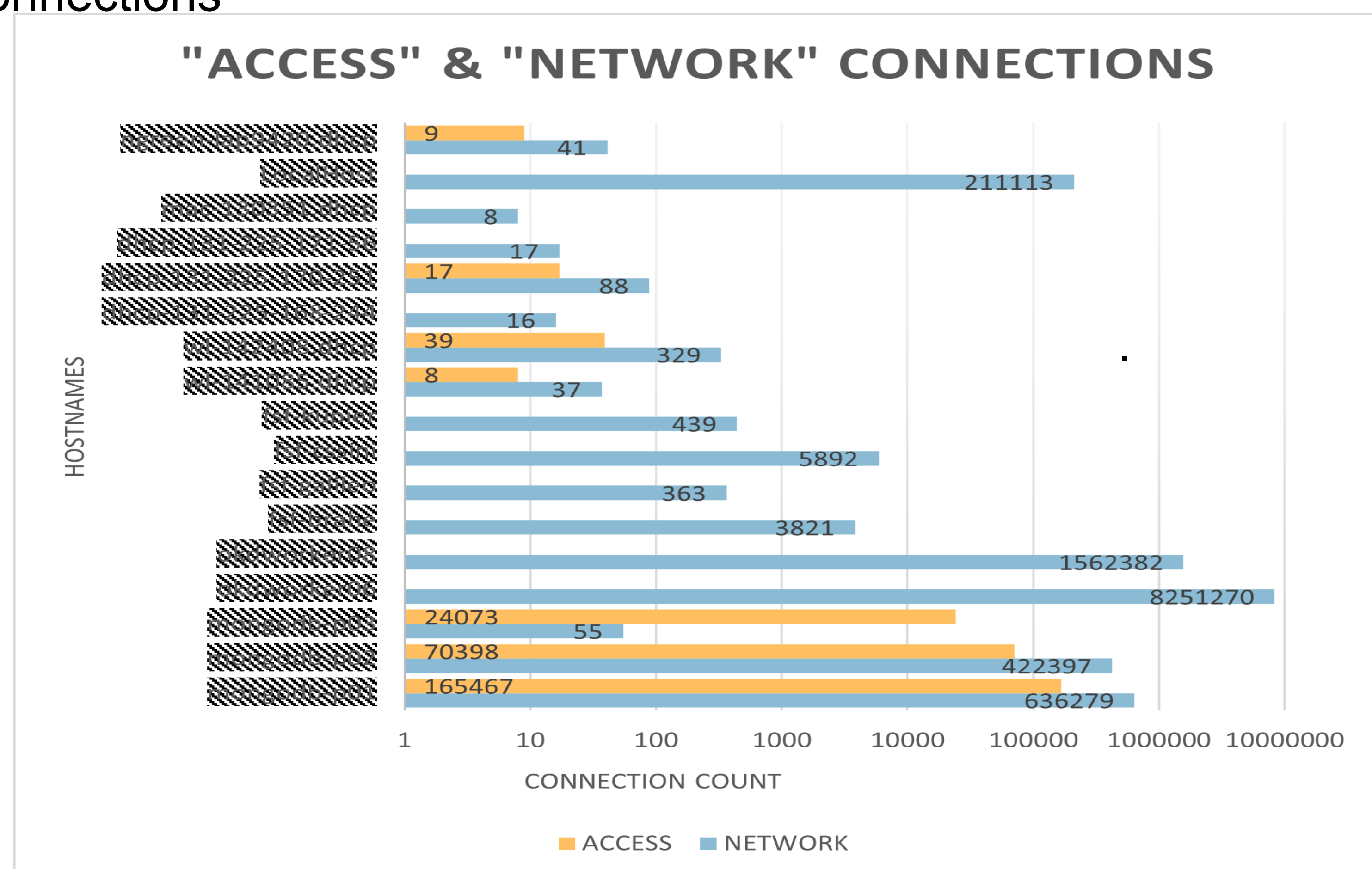


Fig. 3: A graph showing the differences between the numbers of successful connections and attempted connections on a logarithmic scale.

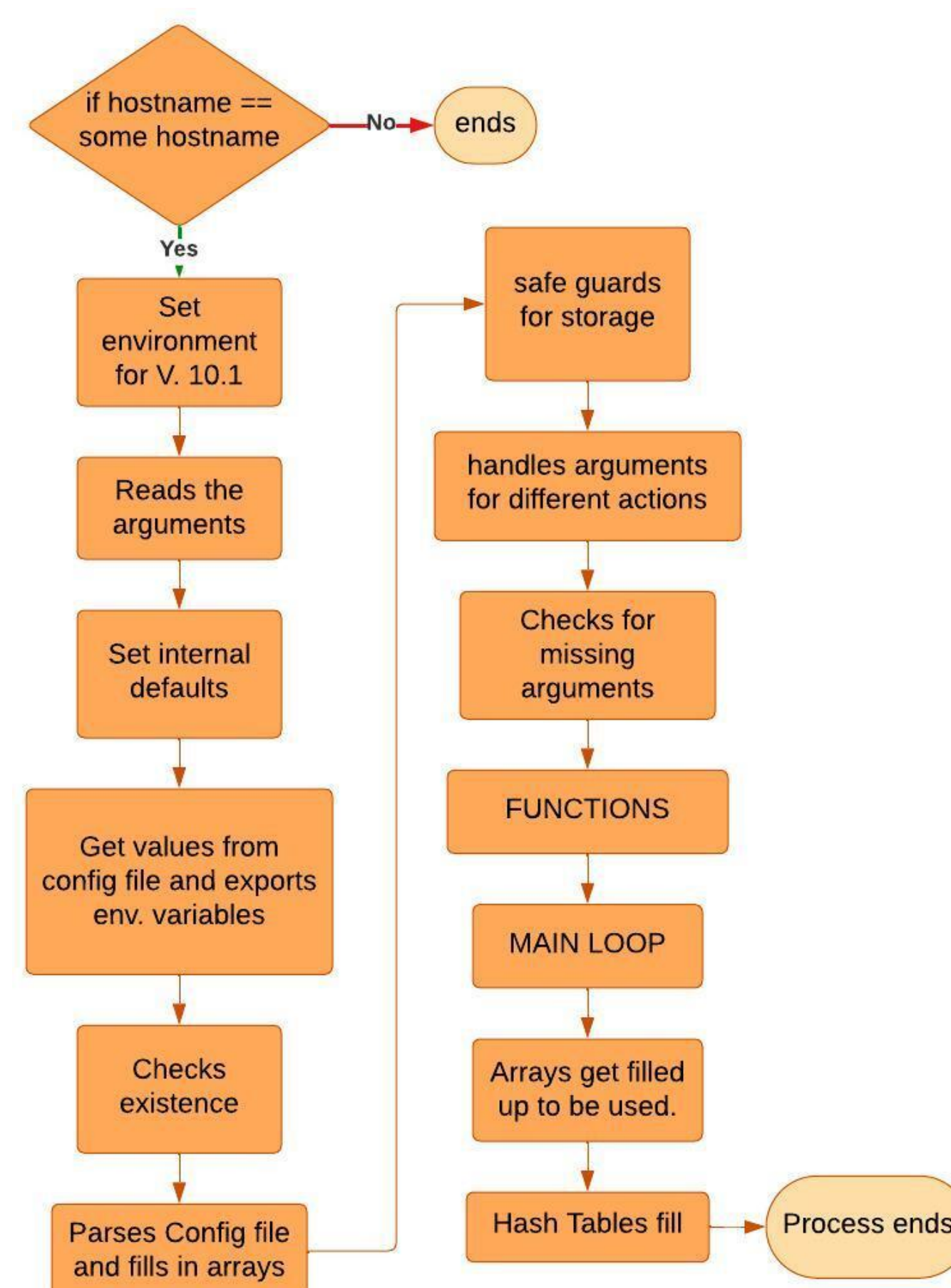


Fig. 1: flow diagram of MyToolMan

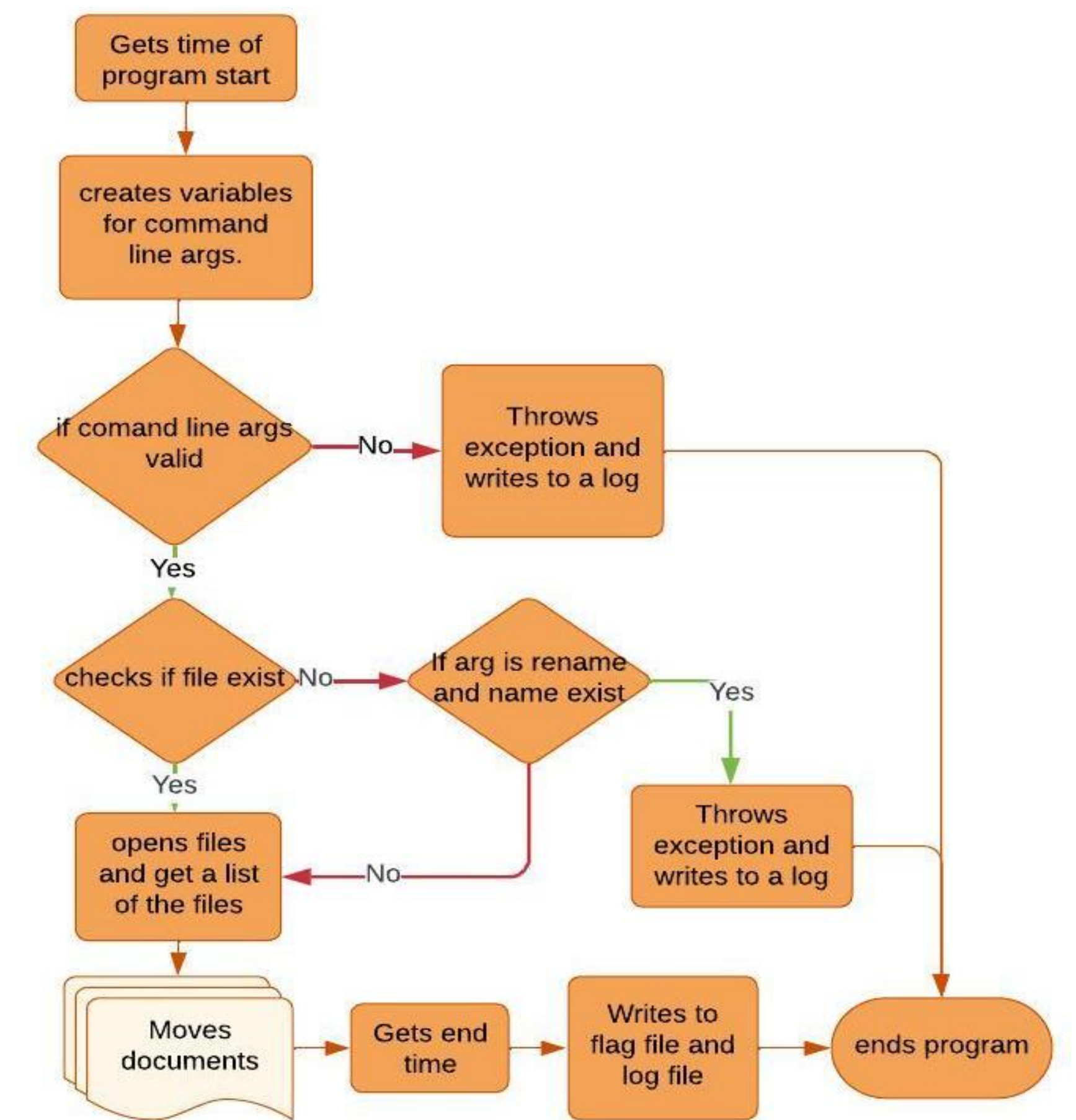


Fig. 2: flow diagram of SQLBackupMover.py.

SQLBackupMover.py Script

For this project I developed a python script that would run on the group's SQL server. These servers already have a backup utility, but it does not have a directory structure that the network storage team can access for their NAS based storage. The script would allow our backups to meet the requirements of the storage team's process

Highlights of the script's main functionality:

- Get the location of backups
- Move the backups to the landing area that the Network storage team process looks for
- Create a flag file for offsite team to inform where the backups are
- Log information about the script's activities

In figure 2 above, you can see the flow of the program. It allows for 3 operations move, copy or rename. It will log the errors as well as when the process was executed, where the new location is, who executed the script, how many files were moved and the total size of the move, in Mb. This script will be implemented on the SQL servers as a post backup process.

Acknowledgments

Thank you to the Database services Group for allowing me to work and learn under their guidance. A special thank you to Natalia Ratnikova for allowing me to work along side you as well as giving me a space to grow my knowledge, and another thank you to Michael Kuc and Olga Vlasova for letting me learning a new programming language to apply to something useful.

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.