Tree Walk Algorithm

Benchmarking ExifTool

- Running the tool on the command line (in seconds):

Number of files in directory	5	10	25	50	100	250	500	1000
Command line execution 1	0.161	0.167	0.164	0.391	0.653	1.503	2.844	5.378
Command line execution 2	0.184	0.124	0.232	0.376	0.74	1.461	2.919	5.135
Command line execution 3	0.118	0.144	0.232	0.39	0.65	1.603	2.89	5.171
Command line execution 4	0.118	0.128	0.242	0.383	0.64	1.495	2.885	5.223
Command line execution 5	0.105	0.141	0.179	0.408	0.647	1.447	2.87	5.259
Average	0.137	0.141	0.210	0.390	0.666	1.502	2.882	5.233

- **Bottleneck:** Small directories
  - Example: 200 Directories with 5 files ~= 200 * 0.137s = 27.4 s

    ➔ 27.4s - 5.24s = **22.16s**

Benchmarking ExifTool

- Executing the tool on 200 directories with 5 files each in one command: 5.788s

➔ Any algorithm will have to combine some amount of the small directories.

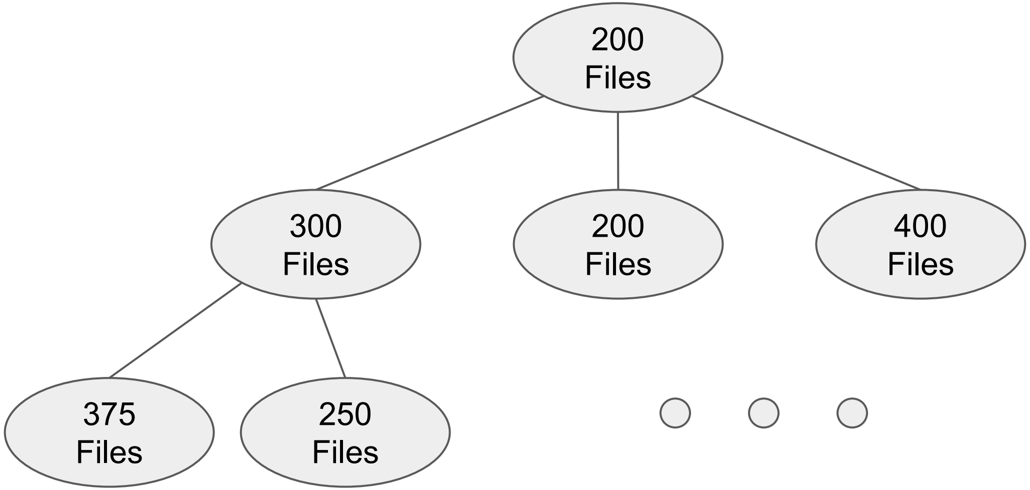
Idea 1: Go directory by directory

- Current approach.
- Deploy the ExifTool directory by directory.
- Possibility to work with multiple threads.
- Incredibly inefficient

Idea 2: Be more efficient with big directories

- Solutions that expects there to be **only directories with a huge amount of files** per directory.
- **Example:**

.



- **TODO:** Find out which approach is more efficient:

1) **Split the threads evenly** among the directories

   Example: 8 directories and 8 threads => Each directory gets 1 thread

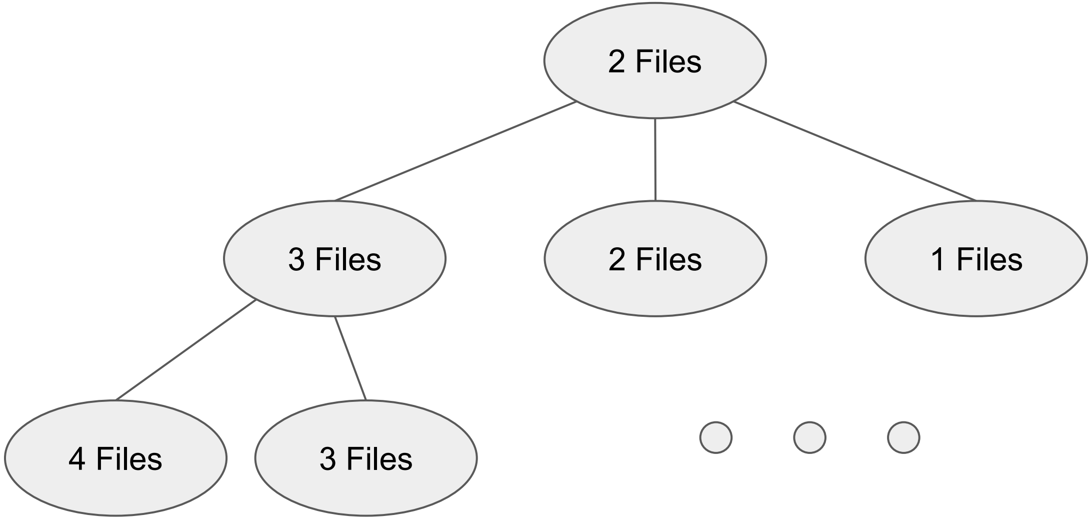
2) Go **directory by directory** and give each the maximum amount of threads.

   ➔ Split the files in X work packages

   Example: 8 directories and 8 threads => Give the first all 8

Idea 3: Be more efficient with small directories

- Solutions that expects there to be only directories with a small amounts of files per directory.
- **Example:**



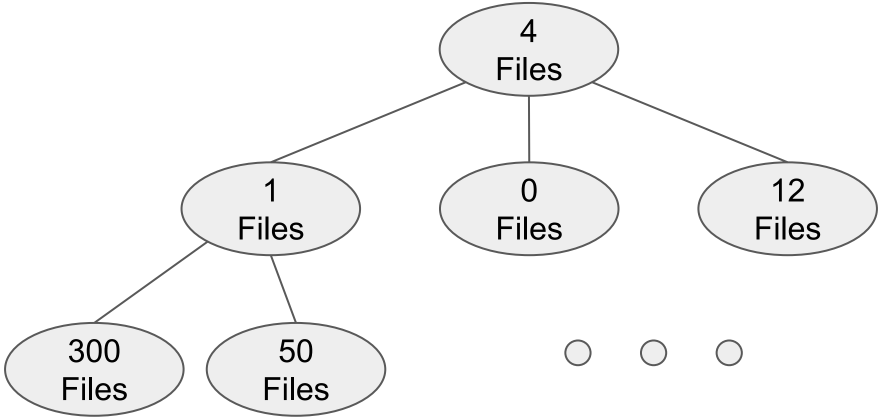
- **TODO:** Find out which approach is more efficient:

1) Go directory by directory and give each directory one thread

2) Try to combine the directories in some way to save ExifTool startup time

Idea 4: Allround Approach

- Solution that expects a huge variance in the amount of files per directory.
- Example:



➔ Try to combine idea 1 and idea 2

- Create a Hash Map that maps directories to the amount of files in the directory.

Key: Number of files

Entry: Directory

- Deploy the ExifTool on the directories with large amount of files first.
    - **TODO:** Determine X
- Use idea 2 for this.
- Either:
    - After all large projects are completely scanned, start working on the small ones.
    - Just combine small directories to be equal to the big directories from the start