

## Design Document

### Assumptions-

- For  $k > 2$ , the subtree which comes first when the first level is sorted alphabetically and is a majority in  $k/2$  file systems is picked.
- If the name, size and SHA1 hash of two files is same the file is assumed to be identical.
- Maximum of 10 file systems.
- Input path to root should be relative to current position and of maximum length 256 characters.

### Structure-

The program has  $k$  way parallelism where  $k$  is the number of file systems. One thread is assigned to traverse each file system. Traversal is done in a BFS manner, wherein all the threads traverse the first level and then wait while the main executes the comparison algorithm. In this, common files are printed as common and common directories are added in a queue. Then each of the threads traverse these directories present in the queue. This process is carried on till we reach the bottom level and the queue becomes empty.

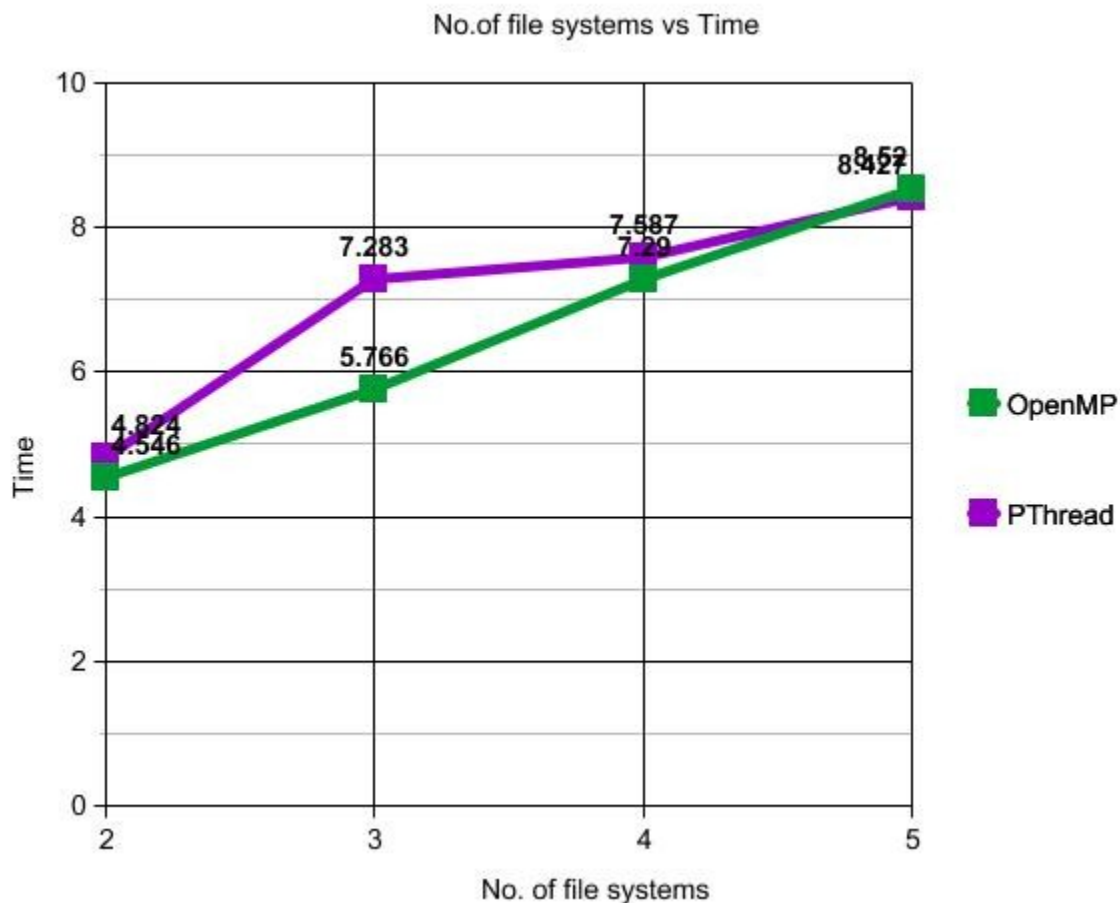
In terms of shared memory, we use a queue to store the next level of directories to be traversed. A two dimensional array of the type struct dirent, where each row corresponds to a file system. Two one dimensional arrays- no\_files: which keeps track of the number of files in a file system at that level and map: which keeps track of the  $k/2$  file systems with the majority subtree in case of  $k > 2$  (map[i]=1 if the file system has the subtree).

### Components-

- 1)omp\_imp.c – driver function for OpenMP implementation.
- 2)pt\_imp.c – driver function for Pthread implementation.
- 3)queue.c – implementation of queue
- 4)thr.c – function which each Pthread executes. It reads from a queue and traverses the directory in the file system it recognizes by an id argument which is passed to it.
- 5)cmp2.c – implementation of compare function in case of 2 file systems. It compares the two sorted array of directory contents and prints which file has been added/deleted if the name of file is different. In case the name of file is same it checks their size and SHA1 hash function. In case any of these is different the file has been edited and if both are same we assume file is identical.
- 6)cmp.c – implementation of compare algorithm in case of more than 2 file systems. It picks the first(alphabetically) directory common in more than  $k/2$  file systems and expands it and prints the common elements in these  $k/2$  file systems.

## Results-

To compare performance of OpenMP vs Pthreads, VampirTrace was used to profile inclusive time taken by main to compare 2, 3, 4 and 5 file systems by both OpenMP and Pthreads. These file system comparison was kept constant when running the two different programmes but in themselves they were of varying depths. The graph obtained has also been send in pdf format in zip folder.



## Discussion-

I discussed the project with the following students-

Gayatri Bhat - 2013A7PS087P

Akhilesh Sudhakar-2013A7PS173P

Lokesh Jain-2012B4A7827P