CS/IS F422 Parallel Computing - Assignment 1

Team : Individual
Marks : 12
Due Date : Sunday, 14th February

Problem Statement:

CompareFS(k=2) Problem:

Consider two purported copies of a filesystem that have been updated independently. Changes may include addition / deletion of file / directory, and change in contents of file.

The given problem is to compare two filesystems (given the pathnames of the roots). The output should be (a) that the two filesystems are exact copies or (b) a list of differences (by pathnames relative to the roots) indicating whether it is addition/deletion or change in contents.

CompareFS(k) Problem:

The second problem is to generalize the comparison to k purported copies of a filesystem, where k is a parameter to your program. The output in this case should be (a) that all filesystems are exact copies or (b) a majority of them have a common subtree of the filesystem (in terms of pathnames and contents) or (c) they are completely divergent i.e. no pathname is common in majority of the copies.

Exercises:

1. Write a program using PThreads to solve the CompareFS(k) problem where a separate thread is used to traverse each filesystem.
2. Write an OpenMP program to solve the CompareFS(k) problem with k-way parallelism.

The exercises should be solved while minimizing the shared memory requirement (also referred to as communication requirement in parallel programming parlance):

- This is best achieved by traversal of filesystems by separate threads in parallel. Individual traversals need not be multi-threaded.
- Exchange of data between threads must be minimal:
  - start by comparing pathnames;
  - if they exist in common (or at least in majority for the k>2 case),
    then for files
    - compare size (in bytes),
    - followed by a signature of the contents (obtained by using any common hashing scheme); and
    - followed by contents

and for directories

- compare list of files (i.e. pathnames).

[Note:
Attempt solving CompareFS(k=2) first and then generalize.
End of Note.]

Deliverables:

(i) Design Document (one or two pages indicating components and their responsibilities, concurrency structure)
(ii) Modular code with brief comments,
(iii) Performance comparison table and plots / graphs for different values of k and different sized filesystems. At least 4 different values of k and for each k at least 4 different filesystem sizes should be tested.

FAQs

1. What is a common copy?
If all the 'k' filesystems have the same file (contentwise, namewise), it is a common copy

2. What is a majority copy?
If ceil(k/2) filesystems have the same file (contentwise, namewise), it is a majority copy

3. How to evaluate performance?
Use VampirTrace profiling tool

4. What details to include in performance comparison table/graph?
Time taken by PThreads/OpenMP programs vs no. of threads, no. of files, depth of filesystem for varying data sizes

5. How do threads communicate?
Figure out how in case of shared memory model – specifically using PThreads and OpenMP.

Fairness Requirement

Students are allowed to discuss the assignment with one another but should report with whom they discussed in the design document. Reported discussion with other students will not be considered as unfair unless such discussion results in the same/similar code.
Read the fairness code in the course handout.