Outlook

Re: "SE3082 – Assignment Proposal

From Nuwan Kodagoda <nuwan.k@sliit.lk>
Date Sat 11/1/2025 12:35 PM
To RATHNAYAKA R M C S it23231528 <it23231528@my.sliit.lk>

This is okay, please proceed

Best Regards

Nuwan

From: RATHNAYAKA R M C S it23231528 <it23231528@my.sliit.lk>
Date: Saturday, 1 November 2025 at 8:52 am
To: Nuwan Kodagoda <nuwan.k@sliit.lk>
Subject: "SE3082 – Assignment Proposal

Dear Mr. Nuwan,

I am Sashanka Rathnayaka (IT23231528) Y3S1 computer science student (SLIIT). I hope you are well. I am writing to submit my algorithm proposal for **Assignment 03** of **SE3082 – Parallel Computing**. Please find the required details below

a) Title of the Algorithm:
"Parallel Graph Traversal Algorithms (BFS and DFS)"

b) Problem Domain:
Sorting and Searching Algorithms – Graph Traversal Algorithms (BFS, DFS)

c) Brief Description (200–300 words):
Graph traversal algorithms such as **Breadth-First Search (BFS)** and **Depth-First Search (DFS)** are widely used to explore and analyze the structure of graphs. BFS explores nodes level by level, using a queue-based approach, while DFS explores nodes along each branch as deeply as possible before backtracking, using recursion or a stack. Both algorithms are essential in computer science applications such as pathfinding, social network analysis, web crawling, and recommendation systems.

These traversal methods are excellent candidates for **parallelization** because the exploration of multiple nodes or adjacency lists can be executed concurrently. For BFS, parallelism can be achieved by processing all vertices at the same level simultaneously, while in DFS, independent recursive paths can be explored concurrently.

The parallel implementations will utilize **OpenMP** (shared-memory threading), **MPI** (distributed node-level parallelization), and **CUDA** (GPU-based traversal). The objective is to compare performance

improvements, scalability, and computational efficiency across these paradigms for both BFS and DFS traversals on large graph datasets

d) Serial C Code (BFS and DFS):

```
#include <stdio.h>

#include <stdlib.h>

#define MAX 100
int visited[MAX], queue[MAX];
int front = -1, rear = -1;

void enqueue(int v) {
   if (rear == MAX - 1) return;
   if (front == -1) front = 0;
   queue[++rear] = v;
}

int dequeue() {
   if (front == -1 || front > rear) return -1;
   return queue[front++];
}

void bfs(int adj[MAX][MAX], int n, int start) {
   for (int i = 0; i < n; i++) visited[i] = 0;
   enqueue(start);
   visited[start] = 1;

   printf("BFS Traversal: ");
   while (front <= rear) {
      int current = dequeue();
      printf("%d ", current);
      for (int i = 0; i < n; i++) {
         if (adj[current][i] == 1 && !visited[i]) {
            enqueue(i);
            visited[i] = 1;
         }
      }
   }
   printf("\n");
}

void dfs(int adj[MAX][MAX], int n, int start) {
   visited[start] = 1;
   printf("%d ", start);
   for (int i = 0; i < n; i++) {
      if (adj[start][i] == 1 && !visited[i]) {
         dfs(adj, n, i);
      }
   }
}
```

```
int main() {
   int n = 5;
   int adj[MAX][MAX] = {
      {0, 1, 1, 0, 0},
      {1, 0, 1, 1, 0},
      {1, 1, 0, 0, 1},
      {0, 1, 0, 0, 1},
      {0, 0, 1, 1, 0}
   };

   printf("Graph Traversals:\n");
   bfs(adj, n, 0);

   for (int i = 0; i < n; i++) visited[i] = 0;
   printf("DFS Traversal: ");
   dfs(adj, n, 0);
   printf("\n");
   return 0;
}
```

Source: From GeeksforGeeks (https://www.geeksforgeeks.org/graph-traversals-bfs-dfs/)

Best regards,
Sashanka Rathnayaka,
IT 23231528.

Disclaimer : This email and any attachment transmitted herewith are confidential and is intended solely for the use of the individual or entity to which they are addressed and may contain information that is privileged or otherwise protected from disclosure. If you are not the intended recipient, you are hereby notified that disclosing, copying, distributing, or taking any action in reliance on this email and the information it contains is strictly prohibited. If you have received this email in error, please notify the sender immediately by reply email and discard all of its contents by deleting this email and the attachment, if any, from your system.

Disclaimer : This email and any attachment transmitted herewith are confidential and is intended solely for the use of the individual or entity to which they are addressed and may contain information that is privileged or otherwise protected from disclosure. If you are not the intended recipient, you are hereby notified that disclosing, copying, distributing, or taking any action in reliance on this email and the information it contains is strictly prohibited. If you have received this email in error, please notify the sender immediately by reply email and discard all of its contents by deleting this email and the attachment, if any, from your system.