King Faisal University
College of Computer Science and IT
Department of Computer Science

Computer Science BS Program

1443/1444H – 2022/2023

CS425 Parallel Computing: Assignment 1, Due date: 28th Sep 2023

Q 1: Select an application of your choice and parallelize its implementation in any programming language (C/C++, Java, and Python). For instance, consider a list of numbers from 1-1000 and iterate on the list and calculate square root of numbers in sequential and parallel execution using Python multiprocessing package. Display the execution time in both cases. (Marks 10)

```python
import math
import time
from multiprocessing import Pool

def calculateSquareRoot(num):
    return math.sqrt(num)

if __name__ == '__main__':
    # Sequential execution
    startTime = time.time()
    numbers = range(1, 1001)
    sequentialResults = [calculateSquareRoot(num) for num in numbers]
    sequentialTime = time.time() - startTime

    # Parallel execution
    startTime = time.time()
    numbers = range(1, 1001)
    with Pool() as pool:
        parallelResults = pool.map(calculateSquareRoot, numbers)
    parallelTime = time.time() - startTime

    print("Sequential Execution Time:", sequentialTime)
    print("Parallel Execution Time:", parallelTime)
```

The output :

```
rs\alija\OneDrive\Documents\Q1.py
Sequential Execution Time: 0.0010232925415039062
Parallel Execution Time: 1.101029872894287
PS C:\Users\alija\OneDrive\Documents>
```

Q 2: Explain what Mandelbrot is and how Mandelbrot set used in computer graphics? Please read the following article and run the code. (Marks 5+5)

https://notebook.community/arasdar/DL/impl-dl/etc/misc3/numba-cuda-gpu-example

Mandelbrot Set:
The Mandelbrot set is a well-known mathematical fractal named after its creator, Benoît B. Mandelbrot. It consists of a group of complex numbers on the complex plane that follow a specific iteration formula. If these numbers remain close to the origin when computed iteratively, they are considered part of the set. However, if they diverge towards infinity, they are not included in the set.

Usage in Computer Graphics:
The Mandelbrot set is widely used in computer graphics to create visually captivating and detailed fractal images. Its applications in this field include:

Image Generation: The Mandelbrot set is employed to generate intricate patterns that are visually appealing. By specifying the boundaries of a complex plane region of interest, the Mandelbrot formula is applied iteratively to each pixel in an image, resulting in the generation of a Mandelbrot set image.

Color Mapping: The color assigned to each pixel in the generated image is determined by the number of iterations needed for the corresponding complex number to escape the Mandelbrot set. Pixels that escape quickly are assigned one color, while those requiring more iterations are assigned different colors. This technique of color mapping enables the creation of vibrant and intricate visual representations of the Mandelbrot set.

Zooming and Exploration: The Mandelbrot set exhibits self-similarity at various levels of detail, making it captivating for exploration. By zooming in on specific regions of the set, one can discover increasingly intricate patterns and structures. This quality makes it a valuable tool for interactive exploration in computer graphics applications.

Parallel Processing: In order to generate high-resolution Mandelbrot images efficiently, parallel processing methods like GPU acceleration can be utilized. These techniques enable faster computations and real-time exploration of the Mandelbrot set.

Q 3: Please read and summarize the following article: (Marks 10)

ZeRO & DeepSpeed: New system optimizations enable training models with over 100 billion parameters

https://www.microsoft.com/en-us/research/blog/zero-deepspeed-new-system-optimizations-enable-training-models-with-over-100-billion-parameters/

Microsoft Research has introduced DeepSpeed, an open-source library compatible with PyTorch, that enhances the training of large natural language models. DeepSpeed incorporates ZeRO (Zero Redundancy Optimizer), a crucial component that optimizes memory usage and enables the training of models with trillions of parameters. ZeRO addresses challenges such as memory limitations by eliminating redundancies across data-parallel processes. It achieves this through three optimization stages: Optimizer State Partitioning, Gradient Partitioning, and Parameter Partitioning. By enabling all three stages, ZeRO can train trillion-parameter models on NVIDIA GPU clusters. The integration of ZeRO into a 17-billion-parameter Turing-NLG model demonstrates improved accuracy and efficiency. DeepSpeed offers benefits in terms of scale, speed, cost, and usability, supporting larger models, increased throughput, reduced training costs, and seamless integration with PyTorch models. The DeepSpeed team welcomes individuals interested in optimizing large-scale system performance to join their efforts.

Note: Upload all answers in the compressed file on BB. Each students has to give the demo of the code in above questions.