Birla Institute of Technology & Science, Pilani
Pilani Campus

II SEMESTER 2022-2023
Assignment

Course No.: CS F422 **Course Title:** Parallel Computing
Deadline: As per Canvas **Maximum Marks:** 40M (20%)

Note:

- This is an individual assignment i.e. maximum of one student per group.
- There are three problems. Any two of three problems need to be submitted.

P1. Implement Blelloch's scan algorithm and Hillis and Steele's algorithm using MPI. You can take a list of numbers in a file "input.txt".

(a) Draw a task dependency graph for the parallel tasks. Identify opportunities for data paralelism, functional parallelism or pipelining. What is degree of concurrency?.
(b) Explain your design in translating the identified parallelism into MPI.
(c) Compute speedup, efficiency, cost and isoefficiency metrics in terms of n, and p where *n* is number of data elements and *p* is number of processors.
(d) Findout whether these algorithms are cost-optimal?
(e) Evaluate the speedup and efficiency by running your program for a range of processes.

Deliverables:

- Design Document (.pdf). Must contain answers for (a)-(e).
- Source code blelloch.c and hillis.c

[20]

P2. Consider a text file (of atleast 1 MB in size) to be encoded using Huffman codes. Now consider parallel algorithms for encoding a given text file and decoding a given encoded file respectively.

(a) Draw a task dependency graph for the parallel tasks. Identify opportunities for data paralelism, functional parallelism or pipelining. What is degree of concurrency?.
(b) Explain your design in translating the identified parallelism into Pthreads.
(c) Compute speedup, efficiency, cost and isoefficiency metrics in terms of n, and p where n is number of data elements and p is number of processors.
(d) Findout whether these algorithms are cost-optimal?
(e) Evaluate the speedup and efficiency by running your program for a range of processes.
(f) Using Pthreads, devise and implement algorithms for encoding and decoding.
(g) Taking any open source serial implementation, apply OpenMP directives for encoiding and decoding.
(h) Compare speedup between (g) and (h)

Deliverables:

- Design Document (.pdf) explaining your design. Must contain answers for (a)-(e) & (h)

- Source code for (f): encode_pthreads.c, decode_pthreads.c
- Source code for (g): encode_openmp.c, decode_openmp.c

[20]

P3. Consider a problem of printing all Kaprekar numbers upto given N taken as CLA. What is Kaprekar number? Consider an n-digit number k. Square it and add the right n digits to the left n or n-1 digits. If the resultant sum is k, then k is called a Kaprekar number. For example, 9 is a Kaprekar number since $9^2 = 81$, and 8 + 1 = 9. $297^2 = 88209$ and 88 + 209 = 297.

(a) Draw a task dependency graph for the parallel tasks. Identify opportunities for data paralelism, functional parallelism or pipelining. What is degree of concurrency?.
(b) Explain your design in translating the identified parallelism into CUDA.
(c) Compute speedup, efficiency, cost and isoefficiency metrics in terms of n, and p where *n* is number of data elements and *p* is number of processors.
(d) Findout whether these algorithms are cost-optimal?
(e) Evaluate the speedup and efficiency by running your program for a range of processes.
(f) Implement your algorithm in CUDA
(g) Taking any open source serial implementation, apply OpenACC directives for computing Kaprekar numbers.
(h) Compare speedup between (g) and (h)

Deliverables:

- Design Document (.pdf). Must contain answers for (a)-(e) & (h).
- Source code kaprekar_cuda.cu
- Source code kaprekar_openacc.c

[20]

--&--