Indian Institute of Technology Guwahati

Department of Computer Science and Engineering

Mid End Semester Exam. Course: CS528 (High Perf. Computing) Date: 27th Feb 2023

Timing: 2.00PM-4.00PM **Full Marks: 50**

(***Write assumption clearly if you assume anything for answering questions***)

==

1. **[10 (=3+3+4) Marks] [Task Scheduling]**

a) Describe the scheduling problem $\boldsymbol{P \mid p_j, pmtn, d_j=D, a_j=0 \mid \sum U_j}$

Ans: Processor envt: m identical/homogeneous processor, Task Envt: All the tasks/jobs are independent tasks with arbitrary execution time and with a common deadline D and arrival/release time 0, pre-emption of job is allowed. The Goal is to minimize number of late jobs ($U_j=1$ if $C_j>D$).

b) Describe the scheduling problem $\boldsymbol{Q \mid p_j, pmtn, prec \mid \sum C_j}$

Ans: Processor envt: m uniform processors (processor with different speed), Task Envt: All the tasks/jobs are with arbitrary execution time and dependent tasks, task pre-emption is allowed and job/tasks are having general precedence dependence among themselves as DAG. The Goal is to minimize summation of Completion time.

c) Solve the above mentioned problem (a) efficiently. Assume $\mathbf{p_j \leq D}$.

Ans: As pre-emption is allowed, all task have same common Deadline and goal is to minimize number of late jobs. Choosing jobs with smaller execution time and trying to schedule with in Deadline D will minimize the sum of late jobs. Approach

- *Sort the job based on SJF and schedule to job in the order to processors if available in the order in any processors ,* ***Fit k smaller tasks such that sum p_j=m.D, So that we can execute using pmtn.***
-
- *This will produce optimal result. And can be proved optimality using exchange argument.*

2. **[10 (=6+4) Marks] [Cilk Program Analysis]**

a) Analyse the following given Cilk code for **Work ($\mathbf{T_1}$)**, **Span ($\mathbf{T_\infty}$)** and **Parallelism**.

```
  cilk void ImageDCT(int **A, int h , int w, int xloc, int yloc) {
    if ( h<thr && w< thr) MatMul(A, B, C, h, w, xloc, yloc) return;
 //Matmul time is O(thr^3) and always get done serially
           spawn  ImageDCT(int **A, h/2 , w/2, xloc+0, yloc+0);
           spawn  ImageDCT(int **A, h/2 , w/2, xloc+h/2, yloc+0);
           spawn  ImageDCT(int **A, h/2 , w/2, xloc+0, yloc+w/2);
           spawn  ImageDCT(int **A, h/2 , w/2, xloc+h/2, yloc+w/2);
           sync;
 }
```

The Cilk function called as ***ImageDCT (A, N, N, 0, 0)*** from main; Assume value of ***N*** and ***thr*** are in integer power of 2 form. Example size of ***N*** is 1024 and ***thr*** is 8.

Ans: Work of the Code: $T_1(N) = 4T_1(N/2)+1$ when N>thr, when N≤ thr $T_1(N)$=thr^3

Amount of work is $O((N/thr)^2)$ from recursion and when we include N≤ thr the T_1 will be $O((N/thr)^2 * thr^3$. So $T_1(N) = O(N^2 * thr)$

Span of the Code : $T_\infty(N) = T_\infty(N/2)+1$ when N>thr, when N≤ thr $T_1(N)$=thr^3, So span $T_\infty(N)$=lg (N/thr)+thr^3=lg N-lg(thr)+thr^3

Parallelism = $T_1(N)/T_\infty(N) = O(N^2 * thr)/$ lg (N/thr)+thr^3

b) Write the assumptions/constraints for work-stealing scheduler to perform optimally. Why random victim section of work-stealing scheduling is good for load-balancing?

Ans: (Part a) Assumptions behind optimality of Work Stealing Scheduler are (a) All the cores/processor are homogeneous, (b) All the memory access takes same amount of time and follows Parallel random access model (PRAM), (c) Communication cost between any two processors are same, which means processor are not grouped in clusters.
Part (b) : Random victim selection in work-stealing proved to be optimal for work propagation from one end to others. Chances for task migration probability is equal from all the corners of the system if victim is selected randomly otherwise the task migration may be skewed. This ensure the proper load balancing of the scheduler.

3. **[10 (=6+4) Marks] [Data Access Optimization]**

a) Given the C-code for polynomial multiplication of two N-degree polynomials, where arrays A and B hold the N coefficients of the input polynoimals and array R to hold resultant 2*N coefficients. We want to accelerate the polynomial multiplication using an accelerator (or cloud system), where the accelerator and the PC do not share the common address space and follows distributed memory model. Calculate the communication complexity and computation complexity of the code. Classify the polynomial application using OPS/Data Transfer approach. Justify your answer about whether it is beneficial to upload the task to the accelerator (or cloud).

```
void Multiply_2Poly(int *A, int *B, int *R, int N){
     for (int i = 0; i<2*N; i++) R[i]=0;
     for (int i=0; i<N; i++)
          for (int j=0; j<N; j++)
               R[i+j] += A[i]*B[j];
}
```

Ans: The above code computes polynomial multiplication. Inputs are two polynomial of N size and output is a polynomial of N size. So the communication/data-transfer complexity is 3N=O(N). Computation Complexity is $O(N^2)$. This application can be classified as OPS/Data Transfer= $O(N^2)/O(N)$ algorithm. Hence it is beneficial to do the polynomial computation using an accelerator/ (cloud uploading).

a) Given the following source code, our aim is to run each iteration of the loop in parallel. Is it possible to run the iterations in parallel? Justify your answer.

```
for(i=0;i<N;i++){
          inX[2*i+3]=X[10*i+2]+(2*i*i+5.0);
}
```

Ans: *There are two array accesses to the array X in the loop 2*i+3 and 10*i+2; we can use GCD test for loop iteration dependence. X[a*i+b]=X[c*i+d], GCD(c,a) divides (d-b) for loop dependence. In this case, a=2, b=3, c=10, d=2; GCD(c,a)=GCD(10,2)=2, d-b=2-3=-1.*
m divide n ==> there exist an integer p such that m.p=n; 2.p=-1 ==>p=-1/2 is not an integer
As 2 does not divide -1. There is no dependence possible.

Others argument that left side access will be always odd and right side access will be even indexes but it not formal[3 marks for this].

4. **[10(=5+5) Marks] [Roop-line Model and Serial Code Opt.]**

a) Given a computer system with peak performance of ***12 TF/s*** and achievable date bandwidth to the compute is ***100 GB/s***. Calculate the expected performance of the following code on the system assuming the size of a float data is ***4B*** and system uses write allocate mode.

```
        for(i=0;i<N;i++){ //float a[N],b[N],c[N],d[N];
                a[i]= s*b[i]+c[i]*d[i];
        }
```

Ans: *There are three read accesses and one write access which also require a read, so in total five access of size 4B. So total memory access amount is 20B for 3 operations.*

*P=min(Pmax, I*bs) = min(12TF/s, 3/20 F/B* 100GB/s)=min(12TF/s, 3*5 GF/s)=*
=min(1200GF/s, 15GF/s)=15 GF/s.

b) Suppose there are ***N*** memories each of latency ***L*** and bandwidth ***B*** are connected to an ***M***-core CPU. Suppose each work accesses ***A*** amount of data/second from randomly chosen memories at a finer granularity.

We aim to minimize the running time of ***W***-independent works using ***T*** threads. Calculate the value of the optimal number of threads to be run in terms of ***L***, ***B***, ***M***, ***N***, ***W*** and ***A***. You may assume the ***T*** threads act as ***T***-workers work simultaneously in executing ***T*** works from the pool, and the execution time of each work may be arbitrary.

Ans: *Tn general memory sub systems are slower. It is desirable that amount of data required by compute sub system should be same as bandwidth provided by memory sub system. So N.B ≈ T.A => So Ans is T=N.B/A.*

- *If $T \geq M$ there is no harm in running more threads on one core.*
- *If N >>T, latency have a small impact as there are N number of memory and as all the accesses are randomly goes to different memories.*

5. [10 (=3+3+4) Marks] [MPI, Amdhal's Law and Computer Network]

a) Suppose a Monte Carlo Simulation software is written using MPI, and the software uses a lot of Scatter and Gather constructs of MPI. Suggest a target interconnection network architecture with minimum interconnection cost for the data centre (HPC System) , such that the Monte-Carlo Simulation application run efficiently.

Ans: *As the App uses a lot of Scatter and Gather construct. Master/Slave architecture is good for system and* ***Star network with master at centre node*** *will have better efficiency with minimum cost as it requires only N-1 links.*

b) Write three possible reasons that may be responsible for achieving superliner speed up ($T_1/T_p = S_p > p$), where ***p*** is the number of processors.

Ans: *Possible reasons behind super linear speed up: (a) Higher memory/cache in case of multiprocessor so that it can fit into memory, (b) multiprocessor architecture is different and improved as compared to single processor one and (c) used single/serial algorithm is not good but parallel algorithm is efficient one.*

c) Suppose ***N*** nodes are connected as a hypercube of stars, where ***sqrt(N)*** number of nodes form a hypercube network and the rest ***[N - sqrt(N)]*** nodes are distributed to each node of the hypercube equally, and all the nodes distributed to a hypercube node are connected as a star network with hypercube node at the center. Calculate bisection, number of links, and diameter of the hypercube of stars. Assume the value of ***N*** is equal to some integer power of 2.

Ans: *Hypercube of Star, Number node is Hypercube is sqrt(N).*

Bisection = sqrt(N)/2
Diameter = lg(sqrt(N))+1+1 : two added for star at hypercube node
*Number of link = #For hypercube + #links Star Part = sqrt(N)*lg(sqrt(N)) + (N-sqrt(N))*