Indian Institute of Technology Guwahati

Department of Computer Science and Engineering

End Semester Examination Course: CS528 (High Performance Computing) Date: 6th May 2022
Timing: 2.00PM-5.00PM (*Model Solution: Answer may differ based on assumption*) **Full Marks: 60**

==

1. [8 (=3+5) Marks] [Basic Scheduling]

a) Describe the problem $P \mid p_j, \text{no-pmtn}, d_j{=}D, a_j{=}0 \mid \sum U_j$

Ans: Scheduling n non-preemptive tasks with a common deadline and arbitrary execution time arrived at time 0 (offline) on m homogeneous processor to minimize number late task.. Late task Uj=1 when fj>D;

b) Solve the above mentioned problem efficiently. Assume $p_j \leq D$.

Ans: As all the tasks arrive at time 0 and have a common deadline D, it is better to schedule the shortest job first (SJF) to maximize number of tasks get fit on m processor before deadline D. This minimizes the number of late jobs

2. [12 (=6+6) Marks] [Reliability and Robustness]

a) Given an application represented using directed tree with N nodes, each node execution time is one unit. There are k nodes in its critical path. All the partial critical paths need to be executed in homogeneous processor/VMs with unit processing speed. To ensure robustness up to three nodes failure and to achieve the minimum C_{max}, calculate the number of VMs required.

Ans: Execution time of node is unit time, k node in the critical path. Robust ness of three node failure, so critical path length changes to k+3. We need to minimize Cmax, and if you assume a PCP can be executed on different VMs, so the number of VM required N/(k+3).

If you assume that a PCP can be executed on only one VM then it will depend on the number of PCP in the directed Tree (with CP=k), the number can be up to N-k PCP in the worst case (Worst case example: k node in CP, rest of the nodes are attached directly to the last node).

b) Given an application with N independent non-pre-emptive tasks and m homogeneous processor but with different failure rates $f_1 \leq f_2 \leq \ldots \leq f_m$. The reliability of task execution is calculated as $exp(-f.t)$, where f is the failure rate of the machine and t is the execution time of the task. System reliability (R_{sys}) is product of reliability of all the tasks. Design an approach to schedule the task in a such a way that it primarily minimizes C_{max} and secondarily maximize the system reliability R_{sys}.

Ans: A : As the primary objective is Cmax, it can be done in two phases

- Phase I : Schedule for Cmax using any good approach using the largest processing time (LPT) rule (or ILP for optimal) assuming all processors with zero failure rate.
- Phase II: Sort the processor index based on load and map the highest loaded processor to the smallest index (processor with lowest failure rate) and so on.

3. [10 (6+4) Marks] [Resource Prediction in Cloud System]

a) Suppose you are using an EWMA predictor $E(t) = \alpha * E(t-1) + (1-\alpha) * O(t)$ with $\alpha = 0.5$, where $E(t)$ and $O(t)$ are estimated and observed values at time t. There is another person, who knows that you use the EWMA model and he/she wants you to make the maximum prediction error and he/she is the person who decides the observed values (between 0 to 100). Assume the initial estimated value is 0. In a long run, what will be the prediction error in percentage?

Ans: In long run it will settle at 2/3*100, which is 66.66% error in prediction.

E(t)	0	50	75	32.5	66.25	33.125	66.xxx	33.xxx	66.xxx	32.xxx
O(t)	100	100	0	**100**	**0**	**100**	**0**	**100**	**0**	**100**
Error	100	50	75	77.5	66.25	72.975	66.xxx	66.xxx	66.xxx	66.xxx

b) Suppose, you figure out that he/she is fooling you, how can you change your strategy to minimize the error? You may switch to another predictor (or change the α value) but the other person still assumes that you are using EWMA prediction with the same α value and he/she continue to pass the observed value based on that.

Ans: Change the predictor based on history of observed values. If observed value is 100 then next time predict 0 and if observed value is zero the predict 100 and continue the same.

4. **[12 (=6+6) Marks] [Resource Consolidation and DVFS]**

a) Given a cloud data centre with m_1 type1 machine, m_2 type2 machines, and total $M\ (= m_1+m_2)$ machines. Power consumption model of type1 and type2 machines are given as $P_{type1} = 200 + 20*u^3$ and $P_{type2} = 50 + 100*u^3$, where u is normalized processor utilization of the machine ($0 < u \leq 1$). There are N webserver tasks (which runs forever) and each task have expected machine utilization u_i (for ith task). Design an approach to map these webserver tasks onto these machines such that total power consumption of the data centre is minimized.

Ans: Calclulation of Pmax and critical utilization

For type 1 machine $P_{type1} = 200 + 20*u^3$. Pmax=220 at u=1, critical utilization cuberoot(200/20*2)=1.7099>1

For type 2 machine $P_{type2} = 50 + 100*u^3$. Pmax=150 at u=1, critical utilization cuberoot(50/2*100)=0.6299<1

- Case I : if sum of utilization of all the tasks **Σu_i < 0.62299*m2**, **run the required number of type2 machine at utilization 0.6299** and tries to fit all the tasks on the machine using any Optimal approach (bin-packing/ILP) if possible. Otherwise increase utilization a bit and tries to fit.
- Case II: if sum of utilization of all the tasks **Σu_i > 0.62299*m2** but **Σu_i < m2**, then tries to fit all the tasks **on the all m2 type2 machines** using any Optimal approach (bin-packing/ILP) by increasing the utilization with step by step starting from 0.6299 upto 1. Motivation here is to save energy by running at lower utilization.
- Case III: if all the type 2 machines are exhausted (Σu_i > m2) than f**irst utilze all the type2 machine and then utilize the type 1 machine, one after another**.

b) Design an optimal approach to solve $P \mid p_j, \textit{no-pmtn}, d_j, a_j{=}0 \mid \sum E_j$, where the power consumption of the processor is modelled as $P = \alpha * f^3$ and number of processor $m = \infty$, E_j is the energy consumption of the task on the processor. Assume $p_j \leq d_j$, $0 < f \leq 1$ and execution time of task on a processor running at frequency f is p_j/f.

Ans: For each task choose a separate processor and run at required frequency $f_j=p_j/d_j$

5. **[10(=4+6) Marks] Roop-line Model and Serial Code Opt.**

a) Given a computer system with peak performance of ***12TF/s*** and achievable date bandwidth to the compute is ***100GB/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++) {a[i]=s*b[i]+c[i]*d[i];} //float a[N],b[N],c[N],d[N];
```

Ans: P_{peak}=12 Tera F/s, I=3F/20B, bs=100GB/s,
Expected Perf.= min(Ppeak,I*bs)= min(12TF/s, 3/20 F/B * 100GB/s)
= **min(12 Tera F/s, 15 GF/s)= 15 GF/s.**

b) Suppose we want to implement an average filter of ***w*** **x** ***w*** size over an Image of size ***N*** **x** ***N*** pixels. Assume ***w*** is odd value and for every pixel position we need to put average of total w^2-1 surrounded pixels and own pixel value. Design an efficient approach to calculate the filtered Image. Analyse the time complexity of your approach in terms of ***N*** and ***w***. You may assume you can use a data type *(similar to Int in Python*) which can store unlimited precision data.

Ans: Suppose input image is X[N][N] and output averaging filtered image is Y[N][N]. Create an another array of I[N][N] which store image integral for each pixel. I[i][j]=$\Sigma_{p=0}^{i} \Sigma_{q=0}^{j}$ X[p][q]. From integral of Image, we can calculate average value for a pixel using two substraction and one addition Y[i][j]=I[i+w/2][j+w/2] – I[i+w/2][j-w/2-1] -I[i-w/2-1][j+w/2] + I[i-w/2-1][j-w/2-1]

This approach will take O(N^2) time to calculate Image Integral and O(N^2) time for average filtering

Ref: https://www.mathworks.com/help/images/ref/integralimage.html

6. **[8 (=4+4) Marks] MPI, Amdhal's Law and Computer Network**

a) Suppose a page ranking software is written in MPI and which has a lot of Map-Reduce constructs and is dominated by many reduce (MPI_reduce) operations. Suggest a target interconnection network architecture of the data center to **efficiently** run the application with minimum interconnection cost.

Ans: Tree interconnect is a natural choice as the Reduce operation can be seen as a tree of operation. Example summation of 4 numbers can be done in tree format as (n1+n2)+(n3+n4), can be done parallelly utilizing links between the tree node in parallel.

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

Ans: Reasons may be responsible

- Parallel version use differnet/efficient algorithm
- Data may fit into cache of multicore, as cache increase with number of processor
- Architecture of multicore may be different/efficient as compared to single processor
- Speed (unit of meseaurement of throughput) may be higher in modern processors as compared to single processor
- Parallel version may have specific to some architetcure communication architetcure which perfectly fit the application scenario