Assignment #2 Solution

CS 5379: Parallel Processing

Name: Ashritha Puradamane Balachandra

R number: R11613952

1. Two statements can execute in parallel if their order of execution does not matter. In other words, if there is no dependency between two sentences, then only they can execute in parallel.

2. Example of output dependence:

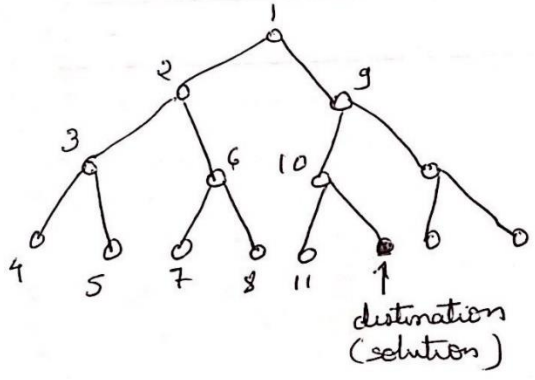
   S1: A = 2 * X
   S2: B = A / 3
   S3: A = 9 * Y

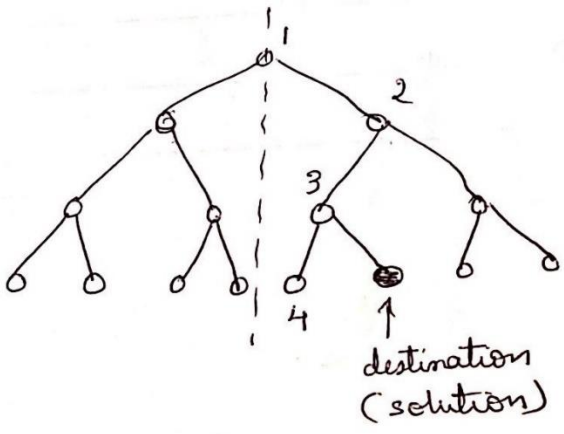
   In the above example, S3 has an output dependence on S1 i.e., $O(S1) \cap O(S3) \neq \emptyset$. S1 has a write and is followed by a write to the same location in S3 (write after write).

   Solution to remove the dependence: Since output dependency is a name dependency, it can be removed through renaming of variables, as in the below modification of the above example.

A = 2 * X	⇨	A2 = 2 * X
B = A / 3		B = A2 /3
A = 9 * Y		A = 9 * Y

3. a.





If a sequential search of the tree is performed using the standard depth-first search (DFS) algorithm, then node traversal will be as shown in the above first diagram. Hence it will **take 11 units of time**.

b. If both processing elements perform a DFS on their respective halves of the tree, then to find the solution it requires **4 units of time**.
The speedup, $S_p = T_s/T_p = 11/4 = 2.75$
There are two processing elements. So, p = 2.
We calculated that speedup, $S_p = 2.75$.
Since parallel system finds the solution in traversing a smaller number of arcs than sequential search (single processing element), it is **super-linear speedup**.

4. a. Gustafson's law id based on fixed-Time Speedup (Scaled Speedup). It emphasis on work finished in a fixed time.
Let problem size is scaled from W to W' and W': Work finished within the fixed time with parallel processing. Then speedup for fixed time is calculated as follows:

$$S'_p = \frac{\text{UniprocessorTime of Solving } W'}{\text{ParallelTimeof Solving } W'}$$

$$S'_p = \frac{\text{UniprocessorTime of Solving } W'}{\text{UniprocessorTimeof Solving } W} = \frac{W'}{W}$$

$$\text{Speedup}_{FT} = \frac{W'}{W} = \frac{\alpha W + (1-\alpha)pW}{W} = \alpha + (1- \alpha)p$$

b. For the following scenarios we want scaled computing:

- When we want to solve larger problems that may not fit or can't run for small machine.
- Scaled computing is needed in a situation when we need better solution, better accuracy.
- We want scaled computing when we need to maintain efficiency.
- In a scenario where we provide real-time solution which may not be achievable with small machines.

5. Scaled computing is desired for the following reasons:

- To solve larger problems which may not fit or can't run for small machine.
- For better solution, better accuracy
- To maintain efficiency
- To provide real-time solution which may not be achievable with small machines.

6. a.

Summary: Since the advent of multi-core technology, an increase in the number of cores per chip gives the more promising future generation of many-core chips. Now a days, research in this area has become one of the hot topics. Cores are able to execute programs or threads independently. Amdahl's argument gives the theoretical speedup in latency of the execution of a task at a fixed workload and it can be used to calculate how much computation can be sped up by running part of it in parallel. According to the modern version of Amdahl's law, its two corollaries are the enhancing a fraction f is small, optimization will also have little effect and the aspect we ignore will limit the speedup. In other words, As speedup S approaches infinity, the speedup is bound by 1/(1-f). Always core resources are increased when sequential performance is greater than the resource of base core equivalents. Otherwise increasing core performance helps sequential execution but it is not the same case with parallel execution. Authors address three possible multicore chips namely symmetric multicore chips, asymmetric multicore chips, and dynamic multicore chips. All symmetric multicore chips will have the same cost for all its cores. These multicore chips support n/r cores of r base core equivalents each.

Problem statement: There exist challenges to the multicore designers to address the chip's overall performance. It is much difficult to answer the questions such as how many cores, which core supposed to use which type of architecture, which type of pipeline design should be used, how to efficiently manage static and dynamic source power. Answer to such questions is not easy to find for the designers as the number of cores per chip increases and hence it seems more challenging in the future. Hardware designers cannot build high-performance multi-cores just by adding more resources and they do not know dynamically add multi-cores for sequential use without affecting the performance and resource overhead.

solution: The authors of this paper offered a corollary of a simple model of multicore hardware resources which provides the view of the entire chip's performance and their implications rather than focusing on core efficiencies to the multicore designers.

Contribution: The proposed model provides insight into discussions and future implications. Authors address three possible multicore chips namely symmetric multicore

chips, asymmetric multicore chips and dynamic multicore chips to provide performance view of the multi-core chips. In symmetric multicore chips, Amdahl's law is applicable because achieving the best speedup requires fs that is close to 1 and using a greater number of base core equivalent can be optimal. Denser chips make the cores to be nonminimal. In the case of asymmetric multicore chips, it can provide much higher speedup value compared to symmetric multicore chips. Denser multicore chips increase both speedup and the optimal performance of the single large core. In dynamic multicore chips, it gives more options for the architects to combine r cores to enhance the performance of only the sequential component. These dynamic multicore chips offer speedups that are greater than symmetric and asymmetric multi-core chips with identical performance functions.

b. **Discussions**: The authors of this paper considered fixed size multi-core chip containing at most n base core equivalents and this restricts the chip designer to give all or a large part of resources to processor cores. The most common misuse assumes that the amount of speedup is independent of the size of the problem. As per Gustafson's law, it is unfair and doesn't give justice to parallel machines because of the time constraints that they allow computations previously done. Explaining the implications of both case of fixed size problem and problem size scaled could have given more clarity for the designers and any parameter that weigh the performance parameter metrics should have given a broader view for the chip designers. Further, nowhere it is mentioned about how the problems scale. This also limits the chip resources on shared cache memory controllers, interconnection networks. It is presumed that few resources are constant in the multicore variations. This could prevent the chip designers from explaining non-processor resources functionality and its impact on overall performance. As the authors mentioned in the paper, since they considered only n base core equivalents, it failed to address the impact on power, area or other factors. In addition, it is assumed that greater sequential performance can be achieved with some techniques for using resources of multiple base core equivalents to create a core. Not considered power as one of the chip's limiting resource. Also, it is equally important to consider structures such as interconnects, shared cache, memory interfaces as the main factors for chip performance contribution. Scheduling software tasks on asymmetric and dynamic multi-core chips may lead to

overhead and it could be difficult. We should note that Amdahl's law is an ideal. In practice, there is the issue of the overhead introduced by parallelizing the code. There are several reasons why some blocks of code can't be parallelized and must be executed in a specific order. It is important to consider such data dependency in the codes while addressing multi-core processors.