University of Central Florida

Department of Computer Science

CDA 5106: Fall 2020

Machine Problem 3: Dynamic Instruction Scheduling

by

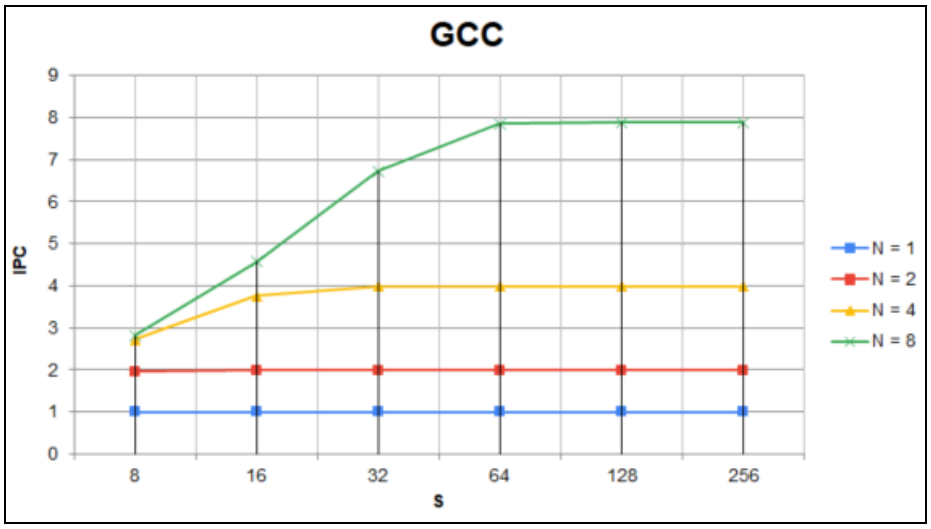
<< Nabila Shahnaz Khan >>

Honor Pledge: "I have neither given nor received unauthorized aid on this test or assignment."

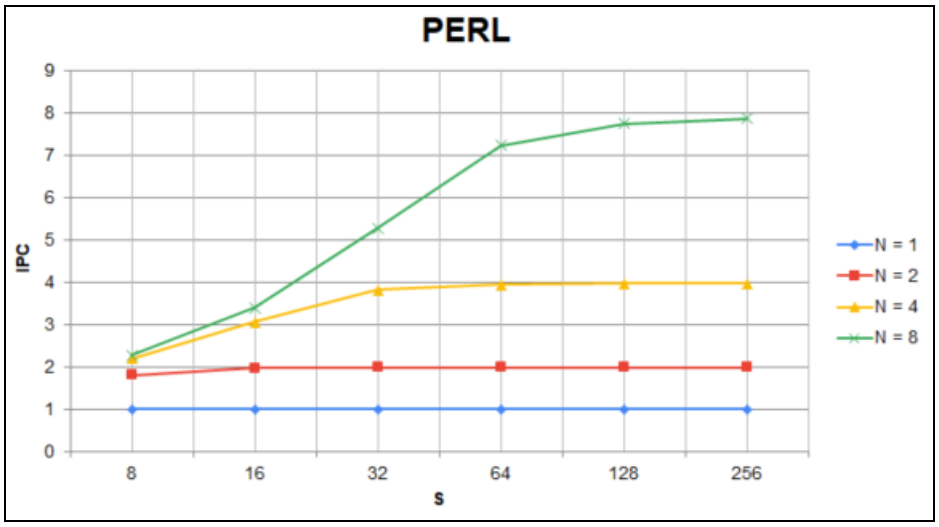
Student's electronic signature: _____Nabila Shahanz Khan_____
(sign by typing your name)

1. Plots containing IPC on the y-axis and Scheduling Queue size(S) on the x-axis are given below. The graph contains four curves with N = 1, 2, 4 and 8. For each N, the Scheduling Queue size(S) are 8, 16, 32, 64, 128 and 256.

Benchmark = gcc



Benchmark = perl



2. Based on above graph, the minimum Scheduling Queue size(S) that still achieves within 5% of the IPC of the largest Scheduling Queue size (256) is given below:

Optimized Scheduling Queue size per peak Fetch Rate		
	Benchmark = gcc	Benchmark = perl
N = 1	8	8
N = 2	8	16
N = 3	32	32
N = 4	64	128

3A. Here, S represents Scheduling Queue size and N represents peak fetch and dispatch rate. The goal of a superscalar processor is to achieve an IPC that is close to peak fetch rate N.

From the graph plots shown above, we can see that when the size of N is too small (ex: N = 1, N = 2), increasing the value of S doesn't have any effect on the IPC (Instruction per Cycle) count. This is because, if less number of instructions is fetched every cycle, a large Scheduling Queue remains idle most of the time. As the value of N increases, the increasing value of S makes a significant difference on the performance as the value of IPC increases with that.

3B. Given different benchmark (trace files) but same microarchitecture configuration, one IPC can be higher or lower than the other. The reason is different benchmarks have different instruction sequence. Also, the operation time of the instructions vary. As result, there's no way to exactly tell if a benchmark will perform better or worse than other benchmarks on a certain microarchitecture configuration. In fact, it is safe to say that benchmarks are microarchitecture configuration independent.