

University of Central Florida

Department of Computer Science

CDA 5106: Fall 2020

Machine Problem 3: Dynamic Instruction Scheduling

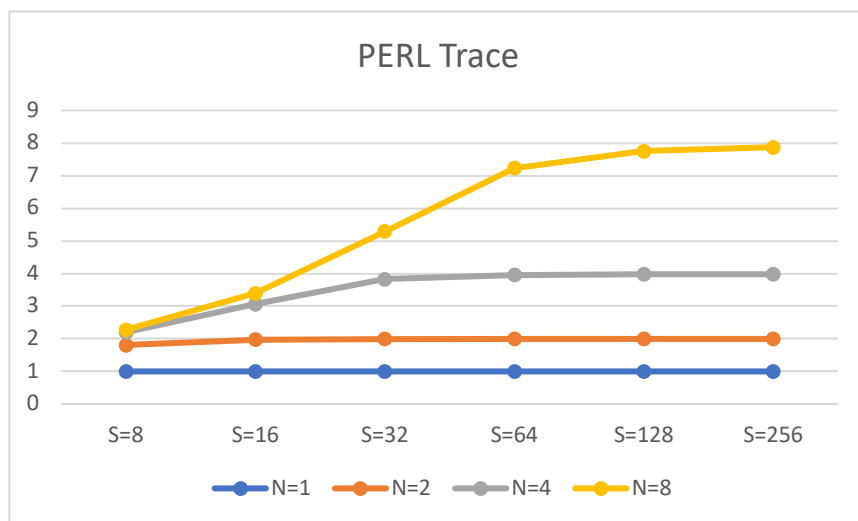
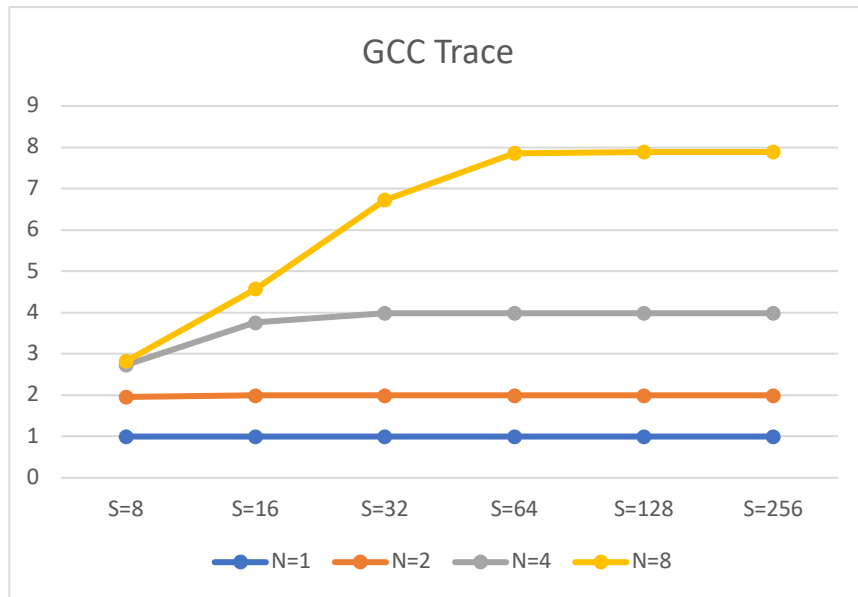
by

<< Alexander Roustai >>

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

Student's electronic signature: _____ Alexander Roustai _____
(sign by typing your name)

Report: Part 1- GCC and PERL graphs



Part 2: Optimized Scheduling Queue size per peak Fetch Rate

	Benchmark = GCC	Benchmark = PERL
N=1	8	8
N=2	8	8
N=4	32	32
N=8	64	128

Part 3:

The goal here is to make a program reach its potential highest performance. N represents the number of pipelines which in a perfect parallel-processing system would mean that an IPC of 8 could occur. The S represents the scheduling queue, which contains the number of reservation stations. In each cycle N instructions are fetched assuming that the reservation stations are not currently full. As S increases, the chances of reservation stations ever becoming full decreases. As S increases, IPC should start to get closer and closer to N. AS your IPC gets closer to N, the number of reservation stations becomes less and less relevant. If we look at our results, we see that the PERL requires more Reservation Stations to complete than the GCC trace. This may be because the PERL trace has operations that require longer to execute, meaning that less reservation stations can become free at a given time, causing the program to stall at times.