NC State University

Department of Electrical and Computer Engineering

ECE 463/563: Fall 2019

Project #3: Dynamic Instruction Scheduling

by

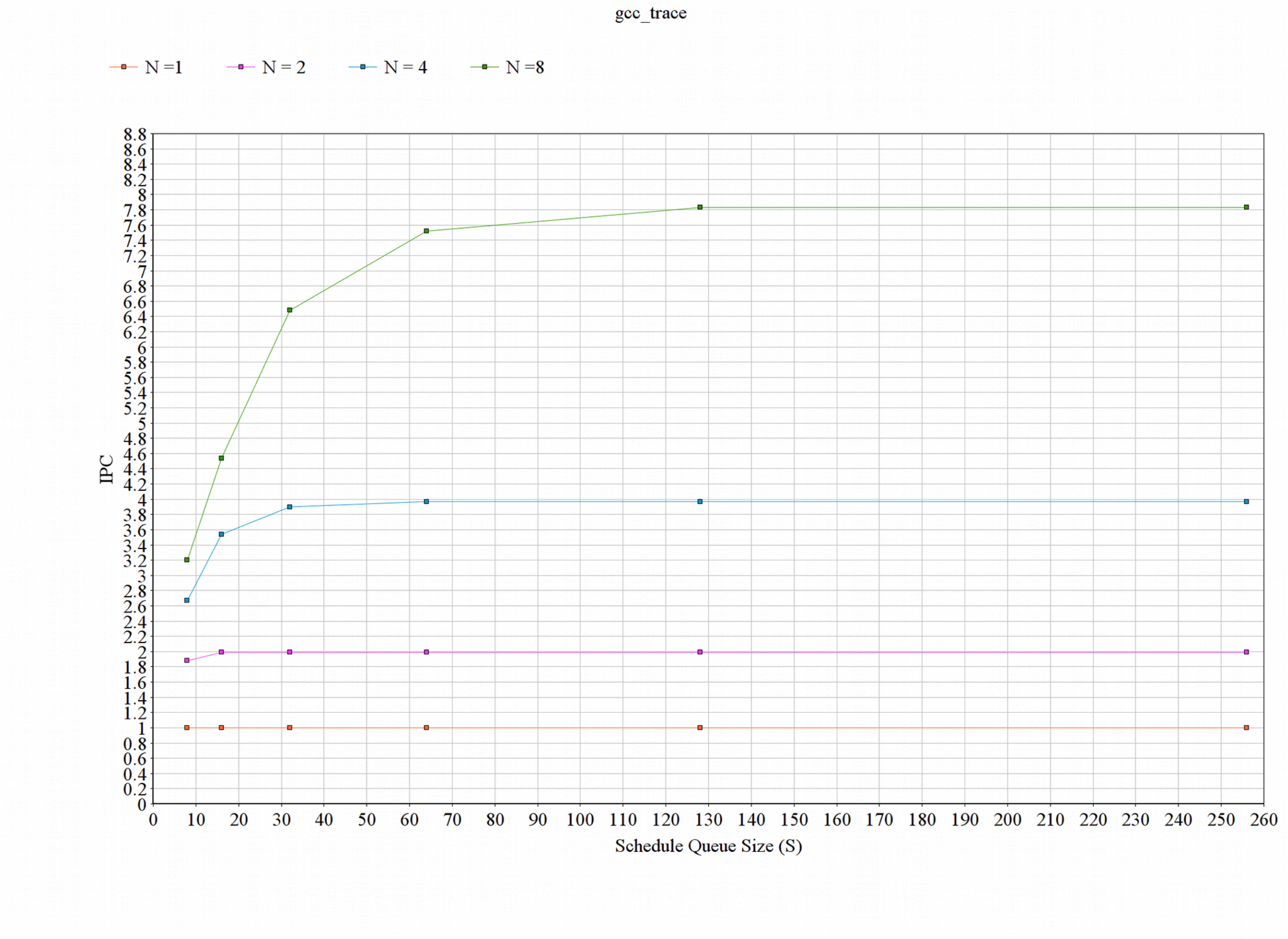
Viswanatha Kasyap Pasumarthy

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

Student’s electronic signature: Viswanatha Kasyap Pasumarthy - 200310870
(sign by typing your name)

Course number: 563
(463 or 563 ?)

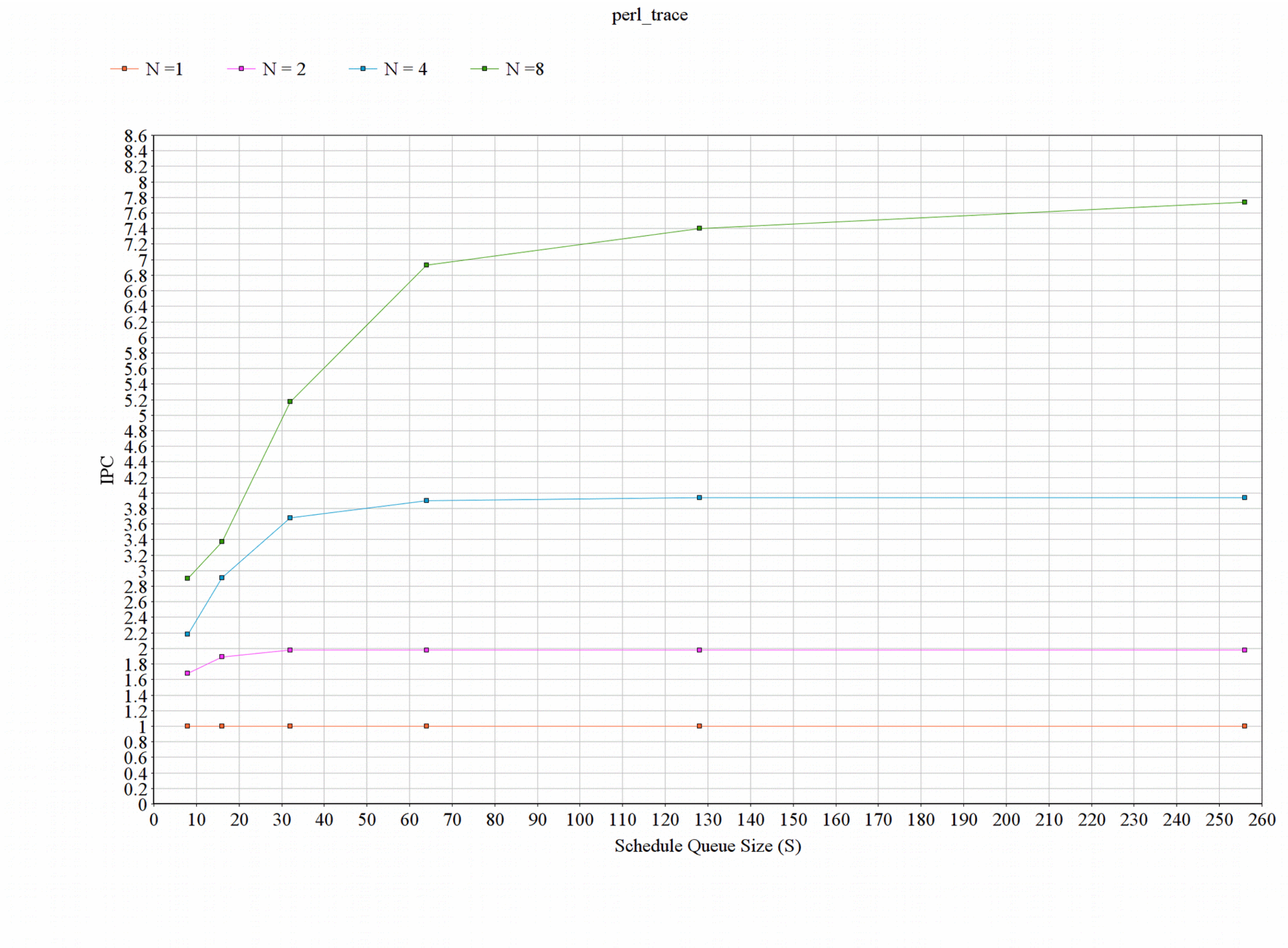
GCC_trace



In this benchmark, we can see that improvements with increase in S are more pronounced with higher values of N. This is because with higher value of N, we can make better use of the larger schedule queue size by moving more instructions at once.

For all values of N, IPC levels out at a certain value of S beyond which adding additional hardware gives no benefits at all. Thus, for a given benchmark, we need to make sure we choose the optimal size of S and N to maximise performance.

PERL_trace



Just like the previous benchmark, here too we see a similar trend between S, N, and IPC.

However, there are slight differences in the rate of improvement which is due to inherent differences in the benchmarks and the number of different types of operations they have.

General Discussions

1. As S increases, we observe an increase in IPC until a certain point beyond which it saturates. The actual value of the IPC itself will depend on the value of N but it is certain that it will level out.

2. As N increases, we observe an unconditional increase in IPC. Regardless of S, if N increases, IPC will go up.

3. In this case, both gcc_trace and perl_trace show similar values of IPC. However, I believe that with different natures of various benchmarks, we will have differing IPC. Some benchmarks may lean heavily toward memory operations while others may not. Thus, it is important that for a given benchmark, we understand the nature of it and choose S and N to optimize it for maximum performance.