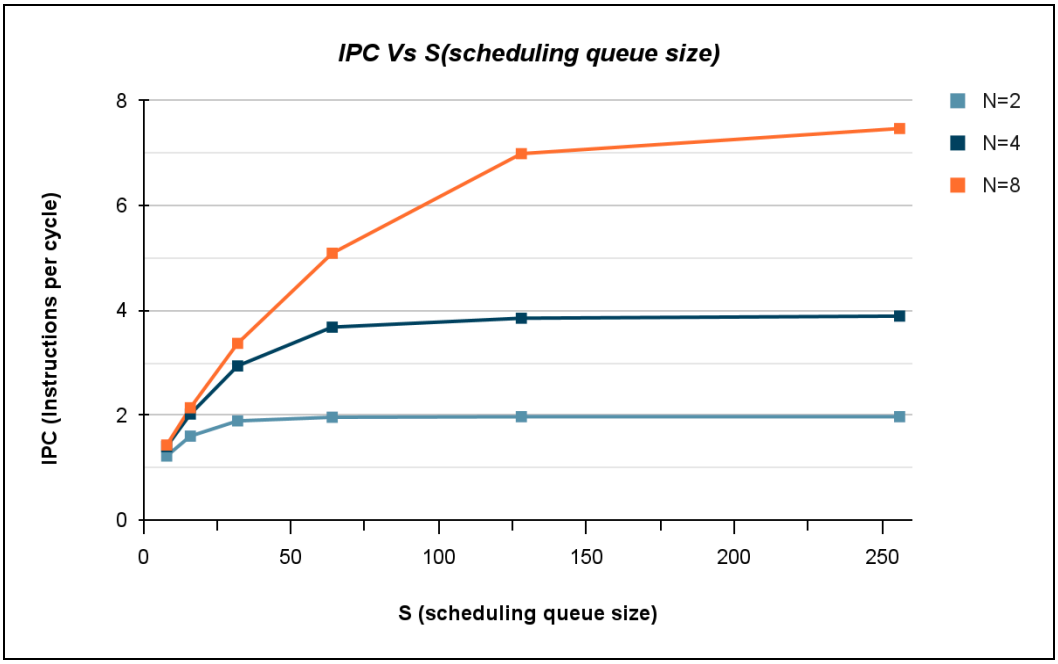
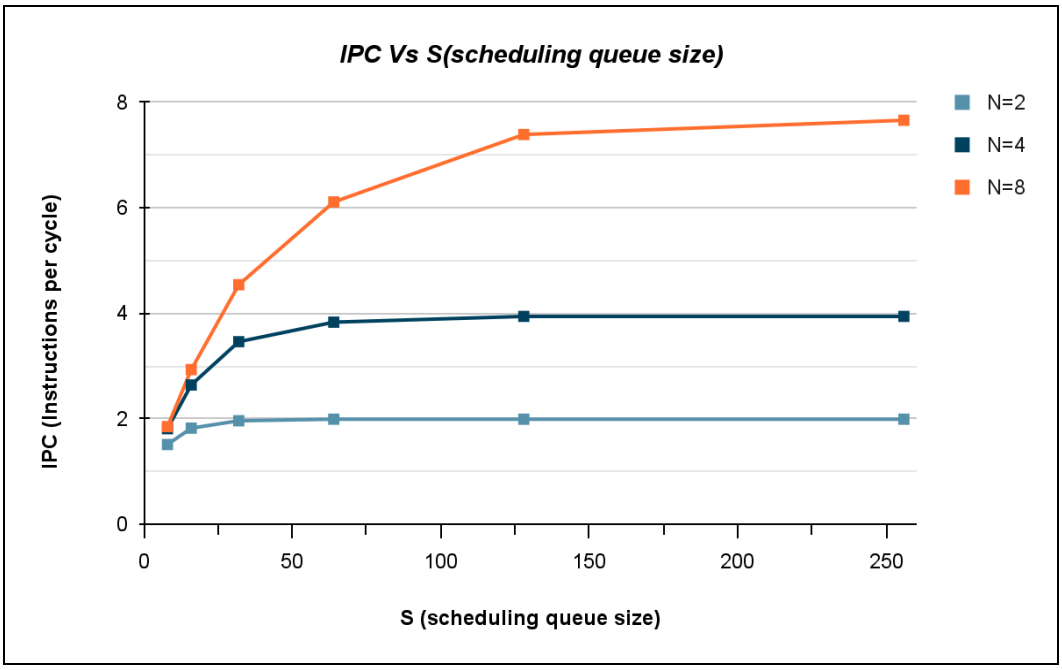
# DYNAMIC SCHEDULING SIMULATOR REPORT

*Plot for perl_trace.txt:*




*Plot for gcc_trace.txt:*



***Trend in IPC as Scheduling queue size(S) increases:***

For both the traces, IPC increases as scheduling queue size increases for any bandwidth of processor. This behaviour is expected because a large scheduling queue provides the processor to issue more instructions simultaneously and pass onto the execute stage, thereby improving the utilization of available functional units and increasing the number of instructions executed per cycle (i.e IPC).

However after a certain point of S, increase in IPC may be diminished. Even though we can have a large number of instructions in the scheduling queue, there could be data dependencies limiting them to pass onto the execute stage. And number of instructions being executed will also be limited by the number of functional units in the execute stage(N). Due to these other factors, increase in IPC is limited after reaching a certain scheduling queue size.

***Trend in IPC as Peak Issue Bandwidth(N) increases:***

For both the traces, IPC increases as N increases and is more when S is also large. This behaviour is expected because larger peak issue bandwidth allows processor to execute more instructions simultaneously using N functional units. But this increase also gets limited by scheduling queue size.

→ We could see that both values of S and N together can play an important role in increasing IPC. Balance of those values are required for better IPC instead of just increasing one of them. Even if one of them is large, IPC gets limited by the other. And IPC is also effected by other factors like data dependencies. So larger N and S may be suitable for high IPC but increasing both of them over a particular point may not effectively lead to a higher IPC.