NC State University

Department of Electrical and Computer Engineering

ECE 463/521: Spring 2012 (Rotenberg)

Project #3: Dynamic Instruction Scheduling

by

<< SALIL KANITKAR >>

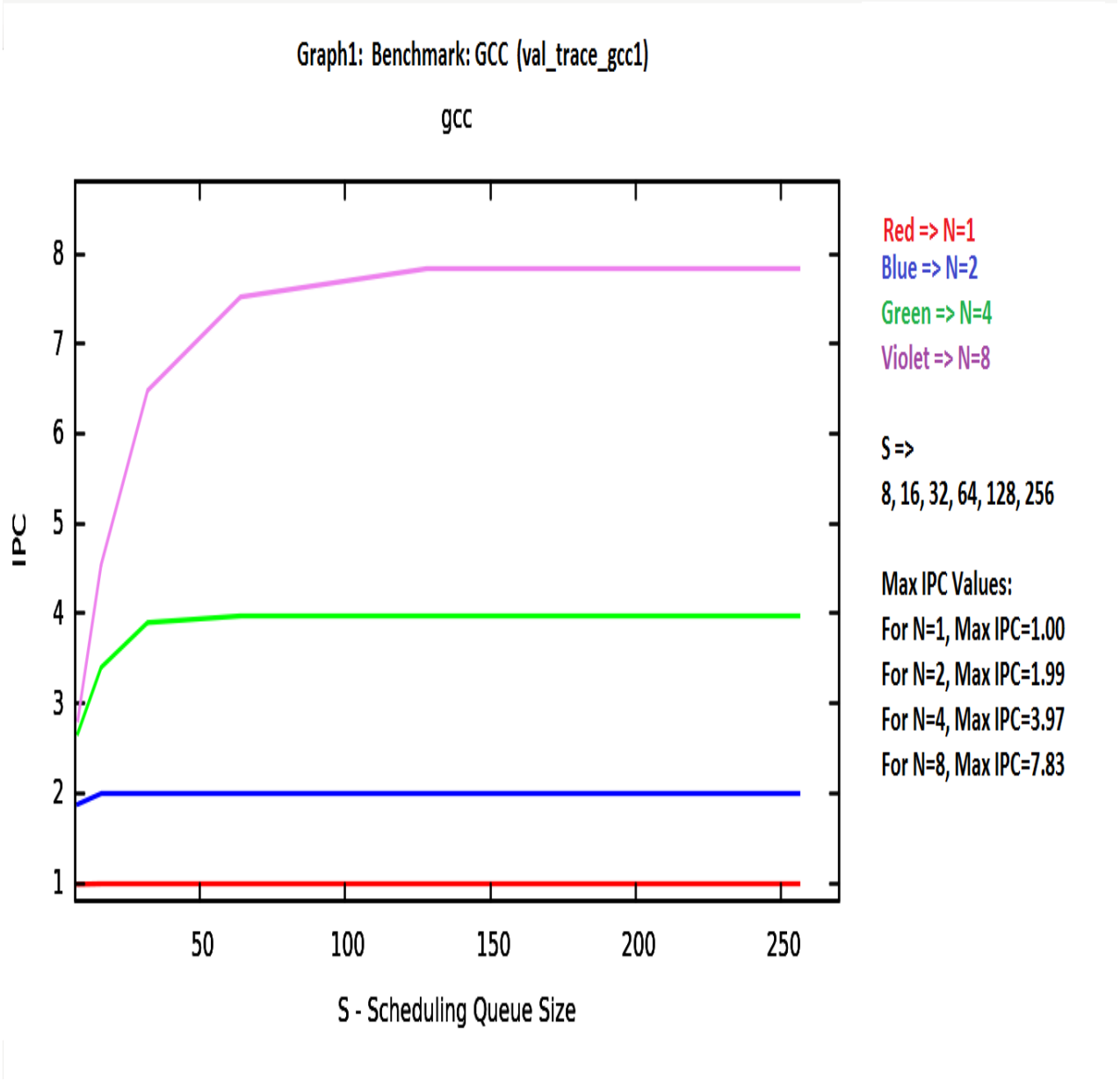
NCSU Honor Pledge: "I have neither given nor received unauthorized aid on this test or assignment."

Student's electronic signature: _____Salil Kanitkar_____
(sign by typing your name)

Course number: _____521_____
(463 or 521 ?)

Report: Project #3. Created 5/5/12.

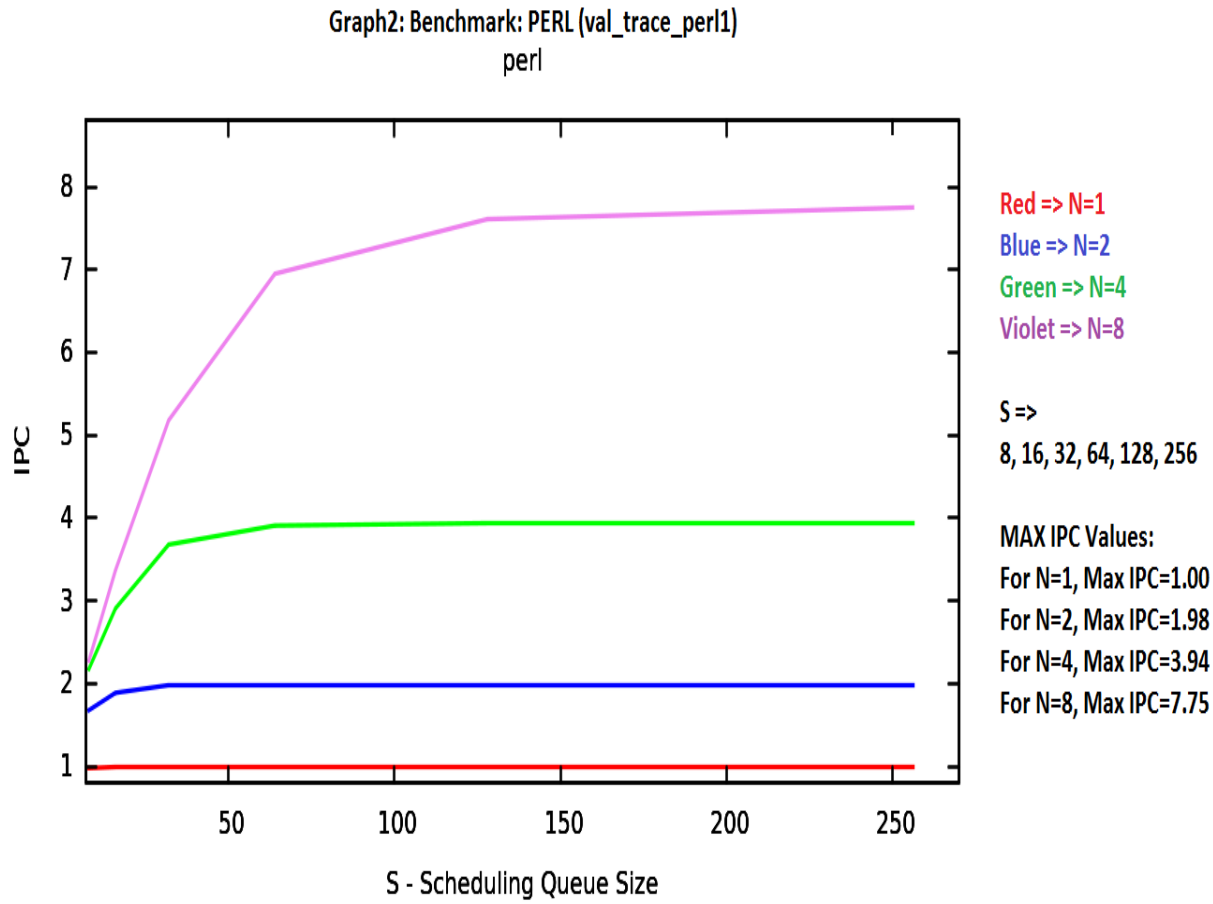
Graph1: gcc Benchmark



Values Used in the Above Plot

	N = 1	**N = 2**	**N = 4**	**N = 8**
S = 8	0.99	1.88	2.67	2.82
S = 16	1.00	1.99	3.4	4.54
S = 32	1.00	1.99	3.90	6.48
S = 64	1.00	1.99	3.97	7.52
S = 128	1.00	1.99	3.97	7.83
S = 256	1.00	1.99	3.97	7.83

Graph2: perl Benchmark



Values Used in the Above Plot

	N = 1	N = 2	N = 4	N = 8
S = 8	0.98	1.68	2.18	2.28
S = 16	1.00	1.89	2.91	3.37
S = 32	1.00	1.98	3.68	5.18
S = 64	1.00	1.98	3.91	6.95
S = 128	1.00	1.98	3.94	7.61
S = 256	1.00	1.98	3.94	7.75

Discussion on Trends in Graphs

Benchmark A: GCC

(1) The **ideal value of IPC (instructions per Cycle) is N** i.e. the Superscalar factor of the microarchitecture configuration. So in the most ideal case, in every cycle, N instructions would complete. This also puts **an upper limit on the value of IPC.**

(2) As the Scheduling Queue Size increases i.e. **as S increases, the IPC increases for given N**. **It goes on increasing till it reaches a saturated value. The saturated value is less than N but very close to N.**

(3) How much close to N the saturated value can be is dependent on the "N".

(4) As N increases, the saturated value of IPC increases in absolute terms, however, the difference between saturated value and N increases, for larger N. For example, for basic case i.e. N, it can be seen that the saturated value is obtained at S=16 only (and of course all the subsequent values of S). However for N=2, the saturated value is **1.99**, for N=4, it is **3.97** and for N=8, it is **7.83**. The difference between saturated value and N is Maximum for larger N.

(5) The saturated value is determined by the level of dependence between the instructions of the benchmark. **Increasing the scheduling queue size for a constant N means that we can look farther into the instruction stream to extract parallelism.**

(6) **For given S, increasing N, increases the IPC value.** This reflects the fact that increasing the superscalar factor provides more fully pipelined functional units – providing more opportunity to exploit parallelism.

Benchmark B: PERL

(1) The **ideal value of IPC (instructions per Cycle) is N** i.e. the Superscalar factor of the microarchitecture configuration. So in the most ideal case, in every cycle, N instructions would complete. This also puts **an upper limit on the value of IPC.**

(2) As the Scheduling Queue Size increases i.e. **as S increases, the IPC increases for given N**. **It goes on increasing till it reaches a saturated value. The saturated value is less than N but very close to N.**

(3) How much close to N the saturated value can be is dependent on the "N".

(4) As N increases, the saturated value of IPC increases in absolute terms, however, the difference between saturated value and N increases, for larger N. For example, for basic case i.e. N, it can be seen that the saturated value is obtained at S=16 only (and of course all the subsequent values of S). However for N=2, the saturated value is **1.98**, for N=4, it is **3.94** and for N=8, it is **7.75**. The difference between saturated value and N is Maximum for larger N.

(5) The saturated value is determined by the level of dependence between the instructions of the benchmark. **Increasing the scheduling queue size for a constant N means that we can look farther into the instruction stream to extract parallelism.**

(6) **For given S, increasing N, increases the IPC value.** This reflects the fact that increasing the superscalar factor provides more fully pipelined functional units – providing more opportunity to exploit parallelism.

Comparison between Benchmarks:

(1) For the **same values of N & S, the IPC value of GCC is greater than PERL** benchmark's IPC value for that configuration.
(2) The reason for GCC having greater IPC value than PERL for the same configuration is that **GCC must be having lesser dependency among instructions as compared to PERL instruction stream.**
(3) **The saturated value for a given S & N configuration is more closer towards N for GCC** than for same S, N combination of PERL.