Objective-B Pointwise Bottleneck Analysis

Harivansh Luchmun
Compiler Construction

May 1, 2025

The four analysis I will feature in this objective is no schedule, and splits with strides 2, 4, and 8

Method used to determine these calculations
R_fe
.objdump - END_INSTRUMENTATION - compute_tst

R_de
.c.mca - uops/ iterations /4

R_ex
.c.osaca - find largest port calc

R_wb
cachegrind - (d1_miss * mem * mem_size)/4

1 Pointwise Operation with no schedule

This bottleneck analysis will be performed on pointwise operation with no schedule

CPU Pipeline Stages	Cycle Count
R_fe	3
R_de	2.75
R_ex	1.20
R_wb	0

Table 1: CPU Pipeline Bottleneck Analysis

2 Pointwise Operation with split stride 2

This bottleneck analysis will be performed on pointwise operation with split schedule of stride 2

CPU Pipeline Stages	Cycle Count
R_fe	3
R_de	3.75
R_ex	1.2
R_wb	0

Table 2: CPU Pipeline Bottleneck Analysis

3 Pointwise Operation with split stride 4

This bottleneck analysis will be performed on pointwise operation with split schedule of stride 4

CPU Pipeline Stages	Cycle Count
R_fe	3
R_de	3.75
R_ex	2,0
R_wb	0

Table 3: CPU Pipeline Bottleneck Analysis

4 Pointwise Operation with split stride 8

This bottleneck analysis will be performed on pointwise operation with split schedule of stride 8

CPU Pipeline Stages	Cycle Count
R_fe	3
R_de	3.75
R_ex	2.0
R_wb	0

Table 4: CPU Pipeline Bottleneck Analysis

Above are the runtimes for the different splits. Split1 is stride 2, split2 is stride 4, spilt3 is stride 8, and none is no split

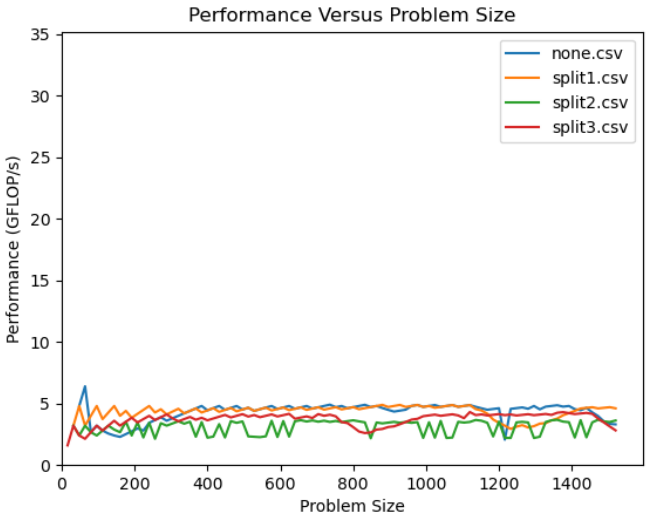

Figure 1: Runtimes with Splits