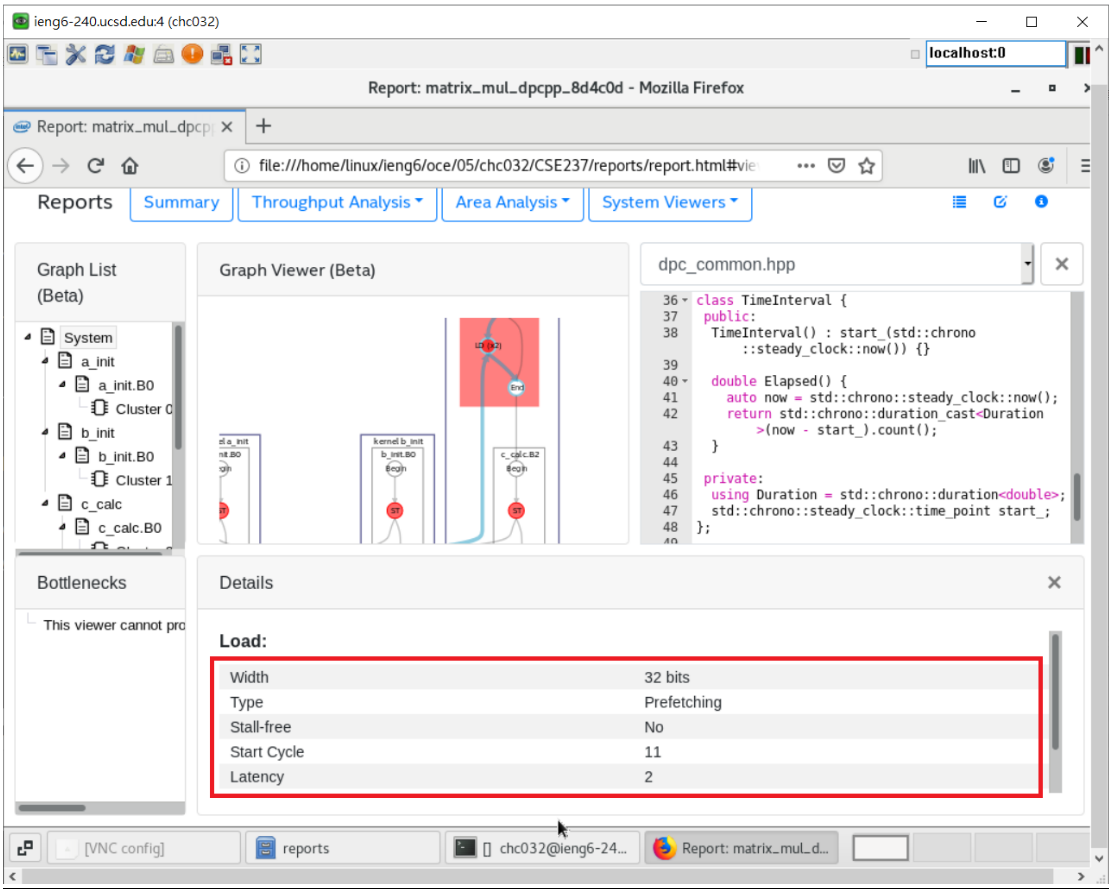
Project4: Matrix Multiplication on Intel DevCloud Using DPC++

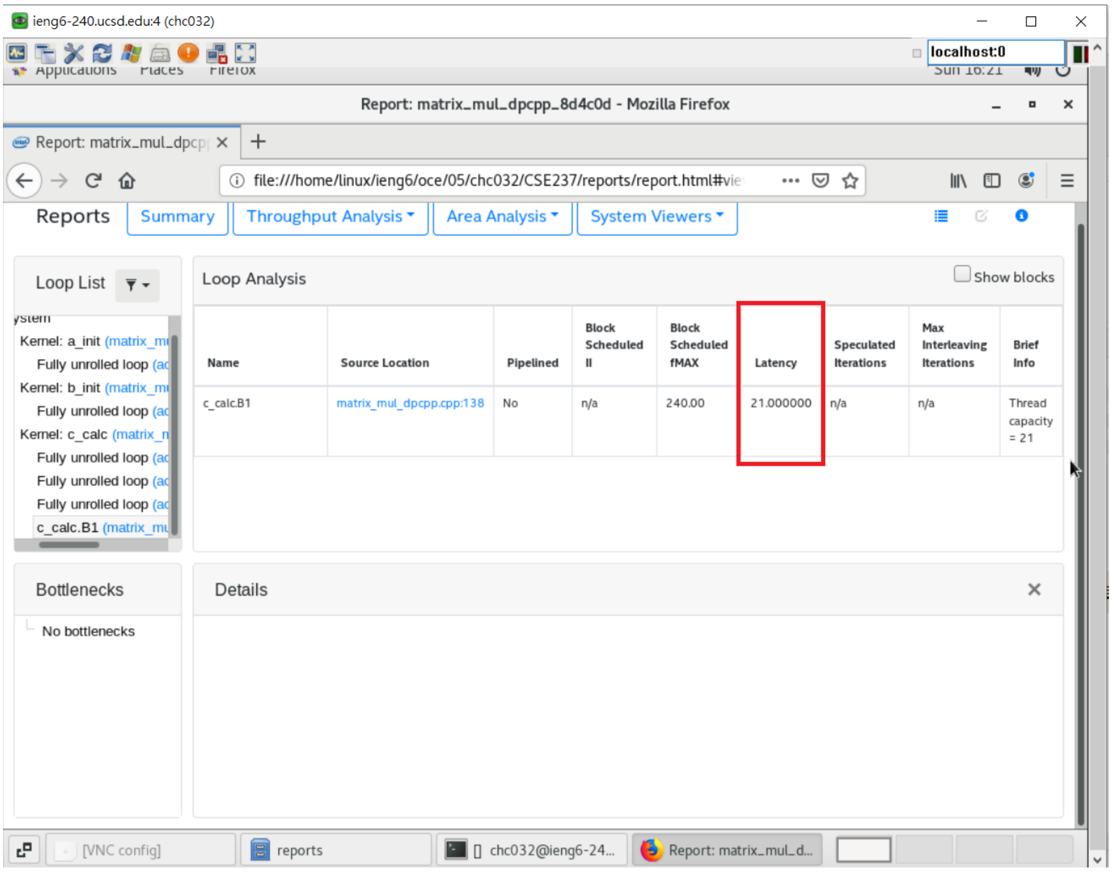
Cheng-Yu Chen

Question 1:

Describe your modification and discuss why it achieves a lower latency.

I change the LSU style from burst-coalesced cached to prefetching. Burst-coalesced cached will buffer contiguous memory requests and wait for the largest burst can be made and then send out the data to the device. Since all of our data are contiguous and the device will have to wait for the buffer reaching largest burst, this method will slow down the whole operation. On the other hand, prefetching method really benefit for contiguous memory reads since it uses FIFO structure which reads large contiguous memory block one time. So, it could read lots of contiguous data one time and it will hasten the circuit. The report did show that the load operation time reduces a lot and the c_calc latency is below 25.





Question 2:

<u>What are the effects and general trends of performing unrolling using the pragma? Are the results as expected?</u>

I set m_size = 144 * 8, which means M = 144, N = 288, P = 576 so a = (144,288), b = (288,576), c = (144,576) and I also use optimized1. I get total latency through this

formula: $\text{Total latency} = \dfrac{\text{number of loop iterations} * \text{reported loop latency}}{\text{unroll\_factor}}$

Factor	2	4	8
Total latency	2160	1656	1404
ALUT	11951	13690	17328
REG	21696	24929	32244
MLAB	193	209	210
RAM	115	167	288
DSP	28	30	34

When the factor increases, all resources increase, too. Especially, ALUT, REG, and RAM increase more aggressively. We could find out that when we unroll the for-loop more, total latency decreases. The results are expected since when the unroll factor increases, the resource utilization increases and the throughput(proportional to the inverse of total latency) also increases and this is the tradeoff between time and area.

Factor = 2:

Loop Analysis

☐ Show blocks

Name	Source Location	Pipelined	Block Scheduled II	Block Scheduled fMAX	Latency	Speculated Iterations	Max Interleaving Iterations	Brief Info
2X Partially unrolled c_calc.B1	matrix_mul_dpcpp.cpp:125	No	n/a	240.00	15.000000	n/a	n/a	Thread capacity = 15

Compile Estimated Kernel Resource Utilization Summary

Name	Source Location	ALM	ALUT	REG	MLAB	RAM	DSP
a_init			2082	4889	49	0	5
b_init			2476	5269	50	0	6
c_calc			4505	7646	94	52	17
Global Interconnect			2888	3825	0	61	0
System description ROM			0	67	0	2	0
Compile Estimated: Kernel System			11951	21696	193	115	28

Factor = 4:

Loop Analysis

☐ Show blocks

Name	Source Location	Pipelined	Block Scheduled II	Block Scheduled fMAX	Latency	Speculated Iterations	Max Interleaving Iterations	Brief Info
4X Partially unrolled c_calc.B1	matrix_mul_dpcpp.cpp:125	No	n/a	240.00	23.000000	n/a	n/a	Thread capacity = 23

Compile Estimated Kernel Resource Utilization Summary

Name	Source Location	ALM	ALUT	REG	MLAB	RAM	DSP
a_init			2082	4889	49	0	5
b_init			2476	5269	50	0	6
c_calc			6234	9600	110	104	19
Global Interconnect			2898	5104	0	61	0
System description ROM			0	67	0	2	0
Compile Estimated: Kernel System			13690	24929	209	167	30

Factor = 8:

Loop Analysis ☐ Show blocks

Name	Source Location	Pipelined	Block Scheduled II	Block Scheduled fMAX	Latency	Speculated Iterations	Max Interleaving Iterations	Brief Info
8X Partially unrolled c_calc.B1	matrix_mul_dpcpp.cpp:125	No	n/a	240.00	39.000000	n/a	n/a	Thread capacity = 39

Compile Estimated Kernel Resource Utilization Summary

Name	Source Location	ALM	ALUT	REG	MLAB	RAM	DSP
a_init			2082	4889	49	0	5
b_init			2476	5269	50	0	6
c_calc			9752	13658	111	225	23
Global Interconnect			3018	8361	0	61	0
System description ROM			0	67	0	2	0
Compile Estimated: Kernel System			17328	32244	210	288	34

Question 3:

What are the effects and general trends of performing manual unrolling? Are the results as expected?

I set m_size = 144 * 8, which means M = 144, N = 288, P = 576 so a = (144,288), b = (288,576), c = (144,576) and I also use optimized1. I get total latency through this formula: $\text{Total latency} = \frac{\text{number of loop iterations} * \text{reported loop latency}}{\text{unroll\_factor}}$

Factor	2	4	8
Total latency	2016	1440	1188
ALUT	11784	13355	16725
REG	21525	24652	31854
MLAB	189	197	214
RAM	115	167	271
DSP	28	30	34

Manual unrolling almost got the same results as question2. When the factor increases, all resources increase, too and ALUT, REG, and RAM increase more aggressively. We could also find out that total latency decreases when unroll factor increase. On the other hand, the unrolling using pragma in question2 performs worse than manual unrolling since it got more latency and use more resources whenever unroll factors are the same. So, manual unrolling could achieve better performance. The results are expected since it shows the tradeoff between time and area.

Factor = 2:

Loop Analysis

Show blocks

Name	Source Location	Pipelined	Block Scheduled II	Block Scheduled fMAX	Latency	Speculated Iterations	Max Interleaving Iterations	Brief Info
c_calc.B1	matrix_mul_dpcpp.cpp:124	No	n/a	240.00	14.000000	n/a	n/a	Thread capacity = 14

Compile Estimated Kernel Resource Utilization Summary

Name	Source Location	ALM	ALUT	REG	MLAB	RAM	DSP
a_init			2082	4889	49	0	5
b_init			2476	5269	50	0	6
c_calc			4338	7475	90	52	17
Global Interconnect			2888	3825	0	61	0
System description ROM			0	67	0	2	0
Compile Estimated: Kernel System			11784	21525	189	115	28

Factor = 4:

Loop Analysis

Show blocks

Name	Source Location	Pipelined	Block Scheduled II	Block Scheduled fMAX	Latency	Speculated Iterations	Max Interleaving Iterations	Brief Info
c_calc.B1	matrix_mul_dpcpp.cpp:124	No	n/a	240.00	20.000000	n/a	n/a	Thread capacity = 20

Compile Estimated Kernel Resource Utilization Summary

Name	Source Location	ALM	ALUT	REG	MLAB	RAM	DSP
a_init			2082	4889	49	0	5
b_init			2476	5269	50	0	6
c_calc			5899	9323	98	104	19
Global Interconnect			2898	5104	0	61	0
System description ROM			0	67	0	2	0
Compile Estimated: Kernel System			13355	24652	197	167	30

Factor = 8:

Loop Analysis

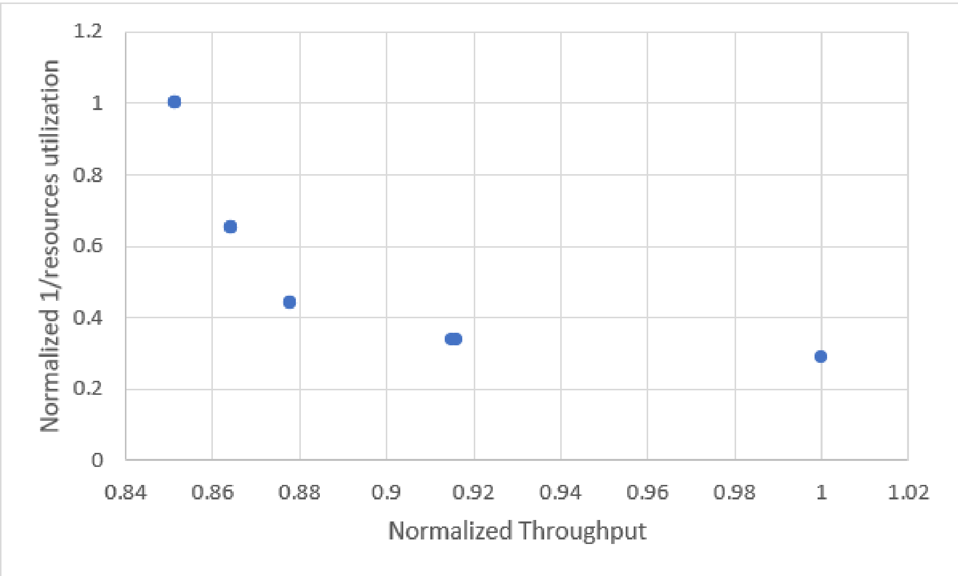
☐ Show blocks

Name	Source Location	Pipelined	Block Scheduled II	Block Scheduled fMAX	Latency	Speculated Iterations	Max Interleaving Iterations	Brief Info
c_calc.B1	matrix_mul_dpcpp.cpp:124	No	n/a	240.00	33.000000	n/a	n/a	Thread capacity = 33

Compile Estimated Kernel Resource Utilization Summary

Name	Source Location	ALM	ALUT	REG	MLAB	RAM	DSP
a_init			2082	4889	49	0	5
b_init			2476	5269	50	0	6
c_calc			9149	13268	115	208	23
Global Interconnect			3018	8361	0	61	0
System description ROM			0	67	0	2	0
Compile Estimated: Kernel System			16725	31854	214	271	34

Question 4:



Total Latency:

Block size→ / Unroll factor↓	4	8	16	32
1	1664	1664	1664	1664
2	1088	1088	1088	1088
4		736	736	736

8			560	560
16				480

Resources:

Block size→ Unroll factor↓	4	8	16	32
1	53611.9	53609.9	53607.9	53604.9
2	54422.9	54420.9	54418.9	54415.9
4		55294.9	55292.9	55289.9
8			57613.9	57688.9
16				62975.9

I assume clock period is the same so the normalized throughput in the above picture would be the inverse of total latency. When the unroll factor increase, the throughput would increase and the resources utilization would increase too. However, when the block size increase, it only reduces very little resources and doesn't change the throughput. On the other hand, when I change the matrix size, I almost get the same result so I don't make a chart of it.