CS6290 Project1 Report

For gcc.trace:

First, I executed the configuration combination of (b1, s1, b2, s2, prefetch policy, replacement policy, and insertion policy) parameters for the gcc.trace.

```
b_values=(5 6 7)
s_values=(0 0 0 0 1 1 1 1 2 2 2 2)
S_values=(2 3 4 5 2 3 4 5 2 3 4 5)
insert_policy=(mip lip)
replacement_policy=(lru lfu)
prefetch_mode=(0 1 2)
trace_files=("gcc.trace")
trace_path="./traces"
for b in "${b_values[@]}"; do
    for i in {0..11}; do
        s=${s_values[i]}
        S=${S_values[i]}
        for trace in "${trace_files[@]}"; do
            for insert in "${insert_policy[@]}"; do
                for r in "${replacement_policy[@]}"; do
                    for p in "${prefetch_mode[@]}"; do
                        ./cachesim -c 15 -b "$b" -s "$s" -C 17 -S "$S" -P "$p" -I "$insert" -r "$r" -f "$trace_path/$trace"
                        echo "Ran: -c 15 -b $b -s $s -C 17 -S $S -P $p -I $insert -r $r -f $trace_path/$trace"
                        sleep 1
                    done
                done
            done
        done
    done
done
```

I prioritized L1 Average Access Time (AAT) in analysis, sorting the results in ascending order.

The table present the top 30 configurations with the lowest L1 AAT for the gcc trace.

C1	B1	S1	C2	B2	S2	Prefetch Policy	Replacement Policy	Insertion Policy	L1_AAT
15	7	2	17	7	4	+1 prefetcher	LRU	LIP.	2.207
15	7	2	17	7	5	+1 prefetcher	LRU	LIP.	2.211
15	7	2	17	7	4	+1 prefetcher	LRU	MIP.	2.217
15	7	2	17	7	5	+1 prefetcher	LRU	MIP.	2.217
15	7	2	17	7	3	+1 prefetcher	LRU	LIP.	2.220
15	7	2	17	7	4	+1 prefetcher	LFU	MIP.	2.230
15	7	2	17	7	4	+1 prefetcher	LFU	LIP.	2.231
15	7	2	17	7	5	+1 prefetcher	LFU	MIP.	2.232
15	7	2	17	7	5	+1 prefetcher	LFU	LIP.	2.232
15	7	2	17	7	3	+1 prefetcher	LRU	MIP.	2.233
15	7	2	17	7	3	+1 prefetcher	LFU	MIP.	2.241
15	7	2	17	7	3	+1 prefetcher	LFU	LIP.	2.242
15	7	2	17	7	5	Strided prefetcher	LRU	LIP.	2.251
15	7	2	17	7	4	Strided prefetcher	LRU	LIP.	2.252
15	7	2	17	7	2	+1 prefetcher	LRU	LIP.	2.256
15	7	2	17	7	2	+1 prefetcher	LFU	MIP.	2.265
15	7	2	17	7	2	+1 prefetcher	LFU	LIP.	2.265
15	7	2	17	7	3	Strided prefetcher	LRU	LIP.	2.269
15	7	2	17	7	5	Strided prefetcher	LFU	LIP.	2.272
15	7	2	17	7	2	+1 prefetcher	LRU	MIP.	2.273
15	7	2	17	7	4	Strided prefetcher	LFU	LIP.	2.274
15	7	2	17	7	5	Prefetcher disabled.	LRU	Prefetcher disabled.	2.274
15	7	2	17	7	5	Strided prefetcher	LFU	MIP.	2.274
15	7	2	17	7	4	Prefetcher disabled.	LRU	Prefetcher disabled.	2.275
15	7	2	17	7	4	Strided prefetcher	LFU	MIP.	2.276
15	7	2	17	7	5	Strided prefetcher	LRU	MIP.	2.283
15	7	2	17	7	3	Prefetcher disabled.	LRU	Prefetcher disabled.	2.287
15	7	2	17	7	3	Strided prefetcher	LFU	LIP.	2.289
15	7	2	17	7	3	Strided prefetcher	LFU	MIP.	2.290
15	7	2	17	7	4	Strided prefetcher	LRU	MIP.	2.296

This result shows that, the B=7 always have better L1_AAT. And I set B to 7 and test more combination of S.

```
b_values=(7)
s_values=(3 3 3 3 4 4 4 4 5 5 5 5)
S_values=(3 4 5 6 4 5 6 7 5 6 7 8)
insert_policy=(mip lip)
replacement_policy=(lru lfu)
prefetch_mode=(0 1 2)
trace_files=("gcc.trace")
trace_path="./traces"
for b in "${b_values[@]}"; do
    for i in {0..11}; do
        s=${s_values[i]}
        S=${S_values[i]}
        for trace in "${trace_files[@]}"; do
            for insert in "${insert_policy[@]}"; do
                for r in "${replacement_policy[@]}"; do
                    for p in "${prefetch_mode[@]}"; do
                        ./cachesim -c 15 -b "$b" -s "$s" -C 17 -S "$S" -P "$p" -I "$insert" -r "$r" -f "$trace_path/$trace"
                        echo "Ran: -c 15 -b $b -s $s -C 17 -S $S -P $p -I $insert -r $r -f $trace_path/$trace"
                        sleep 1
```

```
                    done
                done
            done
        done
    done
done
```

Also, I calculate the Meta_cost and show the Meta cost in the table.

The final result for gcc trace is this:

C1	B1	S1	C2	B2	S2	Prefetch Policy	Replacement Policy	Insertion Policy	Meta_cost	L1_AAT
15	7	4	17	7	4	+1 prefetcher	LRU	LIP.	67328	2.043
15	7	5	17	7	5	+1 prefetcher	LRU	LIP.	68608	2.044
15	7	5	17	7	6	+1 prefetcher	LRU	LIP.	69632	2.044
15	7	4	17	7	5	+1 prefetcher	LRU	LIP.	68352	2.046
15	7	5	17	7	8	+1 prefetcher	LRU	MIP.	71680	2.046
15	7	3	17	7	4	+1 prefetcher	LRU	LIP.	67072	2.047
15	7	5	17	7	7	+1 prefetcher	LRU	LIP.	70656	2.047
15	7	5	17	7	8	+1 prefetcher	LRU	LIP.	71680	2.048
15	7	3	17	7	5	+1 prefetcher	LRU	LIP.	68096	2.049
15	7	4	17	7	6	+1 prefetcher	LRU	LIP.	69376	2.049
15	7	5	17	7	5	+1 prefetcher	LRU	MIP.	68608	2.049
15	7	5	17	7	7	+1 prefetcher	LRU	MIP.	70656	2.049
15	7	5	17	7	6	+1 prefetcher	LRU	MIP.	69632	2.050
15	7	4	17	7	7	+1 prefetcher	LRU	LIP.	70400	2.051
15	7	3	17	7	6	+1 prefetcher	LRU	LIP.	69120	2.052
15	7	4	17	7	5	+1 prefetcher	LRU	MIP.	68352	2.052
15	7	4	17	7	6	+1 prefetcher	LRU	MIP.	69376	2.052
15	7	4	17	7	7	+1 prefetcher	LRU	MIP.	70400	2.052
15	7	4	17	7	4	+1 prefetcher	LRU	MIP.	67328	2.053
15	7	3	17	7	5	+1 prefetcher	LRU	MIP.	68096	2.056
15	7	3	17	7	6	+1 prefetcher	LRU	MIP.	69120	2.056
15	7	3	17	7	3	+1 prefetcher	LRU	LIP.	66048	2.057
15	7	3	17	7	4	+1 prefetcher	LRU	MIP.	67072	2.058
15	7	3	17	7	3	+1 prefetcher	LRU	MIP.	66048	2.070
15	7	5	17	7	5	Strided prefetcher	LRU	LIP.	68608	2.084
15	7	5	17	7	6	Strided prefetcher	LRU	LIP.	69632	2.085
15	7	5	17	7	7	Strided prefetcher	LRU	LIP.	70656	2.086
15	7	5	17	7	8	Strided prefetcher	LRU	LIP.	71680	2.086
15	7	4	17	7	5	Strided prefetcher	LRU	LIP.	68352	2.087
15	7	4	17	7	4	Strided prefetcher	LRU	LIP.	67328	2.088

The selected configuration is as follows:

C1	B1	S1	C2	B2	S2	Prefetch Policy	Replacement Policy	Insertion Policy	Meta_cost	L1_AAT
15	7	4	17	7	4	+1 prefetcher	LRU	LIP.	67328	2.043

For leela.trace:

I run the combination of (b1,s1,b2,s2,prefetch_policy, Replacement_policy and insertion policy) for leela.trace:

```
b_values=(5 6 7)
s_values=(0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5)
S_values=(2 3 4 5 2 3 4 5 2 3 4 5 3 4 5 6 4 5 6 7 5 6 7 8)

insert_policy=(mip lip)
```

```
replacement_policy=(lru lfu)
prefetch_mode=(0 1 2)
trace_files=("leela.trace")
trace_path="./traces"
for b in "${b_values[@]}"; do
    for i in {0..11}; do
        s=${s_values[i]}
        S=${S_values[i]}
        for trace in "${trace_files[@]}"; do
            for insert in "${insert_policy[@]}"; do
                for r in "${replacement_policy[@]}"; do
                    for p in "${prefetch_mode[@]}"; do
                        ./cachesim -c 15 -b "$b" -s "$s" -C 17 -S "$S" -P "$p" -I "$insert" -r "$r" -f "$trace_path/$trace"
                        echo "Ran: -c 15 -b $b -s $s -C 17 -S $S -P $p -I $insert -r $r -f $trace_path/$trace"
                        sleep 1
                    done
                done
            done
        done
    done
done

```

I prioritized L1 Average Access Time (AAT) in analysis, sorting the results in ascending order.

The table present the top 30 configurations with the lowest L1 AAT for Leela trace.

C1	B1	S1	C2	B2	S2	Prefetch Policy	Replacement Policy	Insertion Policy	Meta_cost	L1_AAT
15	7	2	17	7	2	+1 prefetcher	LRU	MIP.	64768	1.952
15	7	2	17	7	2	+1 prefetcher	LFU	MIP.	64768	1.952
15	7	2	17	7	2	+1 prefetcher	LRU	LIP.	64768	1.952
15	7	2	17	7	2	+1 prefetcher	LFU	LIP.	64768	1.952
15	7	2	17	7	3	+1 prefetcher	LRU	MIP.	65792	1.952
15	7	2	17	7	3	+1 prefetcher	LFU	MIP.	65792	1.952
15	7	2	17	7	3	+1 prefetcher	LRU	LIP.	65792	1.952
15	7	2	17	7	3	+1 prefetcher	LFU	LIP.	65792	1.952
15	7	2	17	7	4	+1 prefetcher	LRU	MIP.	66816	1.952
15	7	2	17	7	4	+1 prefetcher	LFU	MIP.	66816	1.952
15	7	2	17	7	4	+1 prefetcher	LRU	LIP.	66816	1.952
15	7	2	17	7	4	+1 prefetcher	LFU	LIP.	66816	1.952
15	7	2	17	7	5	+1 prefetcher	LRU	MIP.	67840	1.952
15	7	2	17	7	5	+1 prefetcher	LFU	MIP.	67840	1.952
15	7	2	17	7	5	+1 prefetcher	LRU	LIP.	67840	1.952
15	7	2	17	7	5	+1 prefetcher	LFU	LIP.	67840	1.952
15	7	2	17	7	2	Strided prefetcher	LRU	MIP.	64768	1.961
15	7	2	17	7	2	Strided prefetcher	LFU	MIP.	64768	1.961
15	7	2	17	7	2	Strided prefetcher	LRU	LIP.	64768	1.961
15	7	2	17	7	2	Strided prefetcher	LFU	LIP.	64768	1.961
15	7	2	17	7	3	Strided prefetcher	LRU	MIP.	65792	1.961
15	7	2	17	7	3	Strided prefetcher	LFU	MIP.	65792	1.961
15	7	2	17	7	3	Strided prefetcher	LRU	LIP.	65792	1.961
15	7	2	17	7	3	Strided prefetcher	LFU	LIP.	65792	1.961
15	7	2	17	7	4	Strided prefetcher	LRU	MIP.	66816	1.961
15	7	2	17	7	4	Strided prefetcher	LFU	MIP.	66816	1.961
15	7	2	17	7	4	Strided prefetcher	LRU	LIP.	66816	1.961
15	7	2	17	7	4	Strided prefetcher	LFU	LIP.	66816	1.961
15	7	2	17	7	5	Strided prefetcher	LRU	MIP.	67840	1.961
15	7	2	17	7	5	Strided prefetcher	LFU	MIP.	67840	1.961

The selected configuration is as follows:

C1	B1	S1	C2	B2	S2	Prefetch Policy	Replacement Policy	Insertion Policy	Meta_cost	L1_AAT
15	7	2	17	7	2	+1 prefetcher	LRU	MIP	64768	1.952

For linpack.trace:

I run the combination of (b1,s1,b2,s2,prefetch_policy, Replacement_policy and insertion policy) for linpack.trace:

```
b_values=(5 6 7)
s_values=(0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5)
S_values=(2 3 4 5 2 3 4 5 2 3 4 5 3 4 5 6 4 5 6 7 5 6 7 8)

insert_policy=(mip lip)
replacement_policy=(lru lfu)
prefetch_mode=(0 1 2)
trace_files=("linpack.trace")
trace_path="./traces"
for b in "${b_values[@]}"; do
    for i in {0..11}; do
        s=${s_values[i]}
        S=${S_values[i]}
        for trace in "${trace_files[@]}"; do
            for insert in "${insert_policy[@]}"; do
                for r in "${replacement_policy[@]}"; do
                    for p in "${prefetch_mode[@]}"; do
                        ./cachesim -c 15 -b "$b" -s "$s" -C 17 -S "$S" -P "$p" -I "$insert" -r "$r" -f "$trace_path/$trace"
                        echo "Ran: -c 15 -b $b -s $s -C 17 -S $S -P $p -I $insert -r $r -f $trace_path/$trace"
                        sleep 1
                    done
                done
            done
        done
    done
done

```

I prioritized L1 Average Access Time (AAT) in analysis, sorting the results in ascending order.

The table present the top 30 configurations with the lowest L1 AAT for linpack trace.

C1	B1	S1	C2	B2	S2	Prefetch Policy	Replace Policy	Insertion Policy	Meta_cost	L1_AAT
15	6	2	17	6	3	Prefetcher disabled.	LRU	Prefetcher disabled.	131584	10.971
15	6	2	17	6	4	Prefetcher disabled.	LRU	Prefetcher disabled.	133632	10.971
15	6	2	17	6	5	Prefetcher disabled.	LRU	Prefetcher disabled.	135680	10.971
15	6	2	17	6	3	Prefetcher disabled.	LFU	Prefetcher disabled.	131584	10.979
15	6	2	17	6	4	Prefetcher disabled.	LFU	Prefetcher disabled.	133632	10.979
15	6	2	17	6	5	Prefetcher disabled.	LFU	Prefetcher disabled.	135680	10.979
15	6	2	17	6	2	Prefetcher disabled.	LRU	Prefetcher disabled.	129536	10.996
15	6	2	17	6	2	Prefetcher disabled.	LFU	Prefetcher disabled.	129536	11.005
15	6	1	17	6	3	Prefetcher disabled.	LRU	Prefetcher disabled.	131072	11.121
15	6	1	17	6	3	Prefetcher disabled.	LFU	Prefetcher disabled.	131072	11.121
15	6	1	17	6	4	Prefetcher disabled.	LRU	Prefetcher disabled.	133120	11.121
15	6	1	17	6	4	Prefetcher disabled.	LFU	Prefetcher disabled.	133120	11.121
15	6	1	17	6	5	Prefetcher disabled.	LRU	Prefetcher disabled.	135168	11.121
15	6	1	17	6	5	Prefetcher disabled.	LFU	Prefetcher disabled.	135168	11.121
15	6	1	17	6	2	Prefetcher disabled.	LRU	Prefetcher disabled.	129024	11.146
15	6	1	17	6	2	Prefetcher disabled.	LFU	Prefetcher disabled.	129024	11.146
15	6	0	17	6	3	Prefetcher disabled.	LRU	Prefetcher disabled.	130560	11.405
15	6	0	17	6	4	Prefetcher disabled.	LRU	Prefetcher disabled.	132608	11.405
15	6	0	17	6	5	Prefetcher disabled.	LRU	Prefetcher disabled.	134656	11.405
15	6	0	17	6	2	Prefetcher disabled.	LRU	Prefetcher disabled.	128512	11.437
15	5	2	17	5	3	+1 prefetcher	LRU	MIP.	263168	11.694
15	5	2	17	5	3	+1 prefetcher	LFU	MIP.	263168	11.694
15	5	2	17	5	3	+1 prefetcher	LRU	LIP.	263168	11.694
15	5	2	17	5	3	+1 prefetcher	LFU	LIP.	263168	11.694
15	5	2	17	5	4	+1 prefetcher	LRU	MIP.	267264	11.694
15	5	2	17	5	4	+1 prefetcher	LFU	MIP.	267264	11.694
15	5	2	17	5	4	+1 prefetcher	LRU	LIP.	267264	11.694
15	5	2	17	5	4	+1 prefetcher	LFU	LIP.	267264	11.694
15	5	2	17	5	5	+1 prefetcher	LRU	MIP.	271360	11.694
15	5	2	17	5	5	+1 prefetcher	LFU	MIP.	271360	11.694

The selected configuration is as follows:

C1	B1	S1	C2	B2	S2	Prefetch Policy	Replacement Policy	Insertion Policy	Meta_cost	L1_AAT
15	6	2	17	6	3	Prefetcher disabled	LRU	Prefetcher disabled	131584	10.971

For matmul_naive.trace:

I run the combination of (b1,s1,b2,s2,prefetch_policy, Replacement_policy and insertion policy) for matmul_naive.trace:

```
b_values=(5 6 7)
s_values=(0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5)
S_values=(2 3 4 5 2 3 4 5 2 3 4 5 3 4 5 6 4 5 6 7 5 6 7 8)

insert_policy=(mip lip)
replacement_policy=(lru lfu)
prefetch_mode=(0 1 2)
trace_files=("matmul_naive.trace")
trace_path="./traces"
for b in "${b_values[@]}"; do
    for i in {0..11}; do
        s=${s_values[i]}
        S=${S_values[i]}
        for trace in "${trace_files[@]}"; do
            for insert in "${insert_policy[@]}"; do
                for r in "${replacement_policy[@]}"; do
```

```
                    for p in "${prefetch_mode[@]}"; do
                        ./cachesim -c 15 -b "$b" -s "$s" -C 17 -S "$S" -P "$p" -I "$insert" -r "$r" -f "$trace_path/$trace"
                        echo "Ran: -c 15 -b $b -s $s -C 17 -S $S -P $p -I $insert -r $r -f $trace_path/$trace"
                        sleep 1
                    done
                done
            done
        done
    done
done

```

I prioritized L1 Average Access Time (AAT) in analysis, sorting the results in ascending order.

The table present the top 30 configurations with the lowest L1 AAT for matmul_naive trace.

C1	B1	S1	C2	B2	S2	Prefetch Policy	Replace Policy	Insertion Policy	Meta_cost	L1_AAT
15	7	2	17	7	3	+1 prefetcher	LFU	MIP.	65792	3.894
15	7	2	17	7	3	+1 prefetcher	LFU	LIP.	65792	3.894
15	7	2	17	7	5	+1 prefetcher	LFU	LIP.	67840	3.900
15	7	2	17	7	5	+1 prefetcher	LFU	MIP.	67840	3.901
15	7	2	17	7	4	+1 prefetcher	LFU	MIP.	66816	3.904
15	7	2	17	7	4	+1 prefetcher	LFU	LIP.	66816	3.905
15	7	2	17	7	3	Strided prefetcher	LFU	LIP.	65792	3.929
15	7	2	17	7	3	Strided prefetcher	LFU	MIP.	65792	3.933
15	7	2	17	7	5	Strided prefetcher	LFU	LIP.	67840	3.934
15	7	2	17	7	5	Strided prefetcher	LFU	MIP.	67840	3.935
15	7	2	17	7	4	Strided prefetcher	LFU	LIP.	66816	3.940
15	7	2	17	7	4	Strided prefetcher	LFU	MIP.	66816	3.942
15	7	2	17	7	2	+1 prefetcher	LFU	MIP.	64768	3.982
15	7	2	17	7	2	+1 prefetcher	LFU	LIP.	64768	3.983
15	7	2	17	7	3	Prefetcher disabled.	LFU	Prefetcher disabled.	65792	3.999
15	7	2	17	7	5	Prefetcher disabled.	LFU	Prefetcher disabled.	67840	4.004
15	7	2	17	7	4	Prefetcher disabled.	LFU	Prefetcher disabled.	66816	4.009
15	7	2	17	7	2	Strided prefetcher	LFU	LIP.	64768	4.018
15	7	2	17	7	2	Strided prefetcher	LFU	MIP.	64768	4.019
15	7	2	17	7	3	+1 prefetcher	LRU	MIP.	65792	4.075
15	7	2	17	7	4	+1 prefetcher	LRU	MIP.	66816	4.075
15	7	2	17	7	5	+1 prefetcher	LRU	MIP.	67840	4.075
15	7	2	17	7	3	+1 prefetcher	LRU	LIP.	65792	4.079
15	7	2	17	7	4	+1 prefetcher	LRU	LIP.	66816	4.080
15	7	2	17	7	5	+1 prefetcher	LRU	LIP.	67840	4.082
15	7	2	17	7	2	Prefetcher disabled.	LFU	Prefetcher disabled.	64768	4.089
15	7	2	17	7	3	Strided prefetcher	LRU	MIP.	65792	4.113
15	7	2	17	7	4	Strided prefetcher	LRU	MIP.	66816	4.113
15	7	2	17	7	5	Strided prefetcher	LRU	MIP.	67840	4.113
15	7	2	17	7	3	Strided prefetcher	LRU	LIP.	65792	4.114

The selected configuration is as follows:

C1	B1	S1	C2	B2	S2	Prefetch Policy	Replacement Policy	Insertion Policy	Meta_cost	L1_AAT
15	7	2	17	7	3	+1 prefetcher	LFU	MIP	65792	3.894

For matmul_tiled.trace

I run the combination of (b1,s1,b2,s2,prefetch_policy, Replacement_policy and insertion policy) for matmul_tiled.trace:

```
b_values=(5 6 7)
s_values=(0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5)
S_values=(2 3 4 5 2 3 4 5 2 3 4 5 3 4 5 6 4 5 6 7 5 6 7 8)

insert_policy=(mip lip)
replacement_policy=(lru lfu)
prefetch_mode=(0 1 2)
trace_files=("matmul_tiled.trace")
trace_path="./traces"
for b in "${b_values[@]}"; do
    for i in {0..11}; do
        s=${s_values[i]}
        S=${S_values[i]}
        for trace in "${trace_files[@]}"; do
            for insert in "${insert_policy[@]}"; do
                for r in "${replacement_policy[@]}"; do
                    for p in "${prefetch_mode[@]}"; do
                        ./cachesim -c 15 -b "$b" -s "$s" -C 17 -S "$S" -P "$p" -I "$insert" -r "$r" -f "$trace_path/$trace"
                        echo "Ran: -c 15 -b $b -s $s -C 17 -S $S -P $p -I $insert -r $r -f $trace_path/$trace"
                        sleep 1
                    done
                done
            done
        done
    done
done

```

I prioritized L1 Average Access Time (AAT) in analysis, sorting the results in ascending order.

The table present the top 30 configurations with the lowest L1 AAT for matmul_tiled trace.

C1	B1	S1	C2	B2	S2	Prefetch Policy	Replace Policy	Insertion Policy	Meta_cost	L1_AAT
15	7	2	17	7	3	+1 prefetcher	LRU	MIP.	65792	2.059
15	7	2	17	7	4	+1 prefetcher	LRU	MIP.	66816	2.059
15	7	2	17	7	5	+1 prefetcher	LRU	MIP.	67840	2.059
15	7	2	17	7	2	+1 prefetcher	LRU	MIP.	64768	2.060
15	7	2	17	7	5	+1 prefetcher	LRU	LIP.	67840	2.061
15	7	2	17	7	3	+1 prefetcher	LRU	LIP.	65792	2.062
15	7	2	17	7	4	+1 prefetcher	LRU	LIP.	66816	2.062
15	7	2	17	7	2	+1 prefetcher	LRU	LIP.	64768	2.063
15	7	2	17	7	4	+1 prefetcher	LFU	MIP.	66816	2.067
15	7	2	17	7	5	+1 prefetcher	LFU	MIP.	67840	2.067
15	7	2	17	7	5	+1 prefetcher	LFU	LIP.	67840	2.067
15	7	2	17	7	4	+1 prefetcher	LFU	LIP.	66816	2.068
15	7	2	17	7	3	+1 prefetcher	LFU	MIP.	65792	2.070
15	7	2	17	7	3	+1 prefetcher	LFU	LIP.	65792	2.070
15	7	2	17	7	2	+1 prefetcher	LFU	MIP.	64768	2.072
15	7	2	17	7	2	+1 prefetcher	LFU	LIP.	64768	2.073
15	7	2	17	7	5	Strided prefetcher	LRU	MIP.	67840	2.089
15	7	2	17	7	3	Strided prefetcher	LRU	MIP.	65792	2.090
15	7	2	17	7	3	Strided prefetcher	LRU	LIP.	65792	2.090
15	7	2	17	7	4	Strided prefetcher	LRU	MIP.	66816	2.090
15	7	2	17	7	4	Strided prefetcher	LRU	LIP.	66816	2.090
15	7	2	17	7	5	Strided prefetcher	LRU	LIP.	67840	2.090
15	7	2	17	7	2	Strided prefetcher	LRU	LIP.	64768	2.091
15	7	2	17	7	2	Strided prefetcher	LRU	MIP.	64768	2.093
15	7	2	17	7	4	Strided prefetcher	LFU	MIP.	66816	2.096
15	7	2	17	7	4	Strided prefetcher	LFU	LIP.	66816	2.096
15	7	2	17	7	5	Strided prefetcher	LFU	MIP.	67840	2.096
15	7	2	17	7	5	Strided prefetcher	LFU	LIP.	67840	2.096
15	7	2	17	7	3	Strided prefetcher	LFU	MIP.	65792	2.098
15	7	2	17	7	3	Strided prefetcher	LFU	LIP.	65792	2.098

The selected configuration is as follows:

C1	B1	S1	C2	B2	S2	Prefetch Policy	Replacement Policy	Insertion Policy	Meta_cost	L1_AAT
15	7	2	17	7	3	+1 prefetcher	LRU	MIP	65792	2.059

For mcf.trace:

I run the combination of (b1,s1,b2,s2,prefetch_policy, Replacement_policy and insertion policy) for mcf trace:

```
b_values=(5 6 7)
s_values=(0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5)
S_values=(2 3 4 5 2 3 4 5 2 3 4 5 3 4 5 6 4 5 6 7 5 6 7 8)

insert_policy=(mip lip)
replacement_policy=(lru lfu)
prefetch_mode=(0 1 2)
trace_files=("mcf.trace")
trace_path="./traces"
for b in "${b_values[@]}"; do
    for i in {0..11}; do
        s=${s_values[i]}
        S=${S_values[i]}
        for trace in "${trace_files[@]}"; do
            for insert in "${insert_policy[@]}"; do
```

```
                for r in "${replacement_policy[@]}"; do
                    for p in "${prefetch_mode[@]}"; do
                        ./cachesim -c 15 -b "$b" -s "$s" -C 17 -S "$S" -P "$p" -I "$insert" -r "$r" -f "$trace_path/$trace"
                        echo "Ran: -c 15 -b $b -s $s -C 17 -S $S -P $p -I $insert -r $r -f $trace_path/$trace"
                        sleep 1
                    done
                done
            done
        done
    done
done

```

I prioritized L1 Average Access Time (AAT) in analysis, sorting the results in ascending order.

The table present the top 30 configurations with the lowest L1 AAT for mcf trace.

C1	B1	S1	C2	B2	S2	Prefetch Policy	Replace Policy	Insertion Policy	Meta_cost	L1_AAT
15	7	2	17	7	5	+1 prefetcher	LRU	MIP.	67840	2.070
15	7	2	17	7	4	+1 prefetcher	LRU	MIP.	66816	2.071
15	7	2	17	7	3	+1 prefetcher	LRU	MIP.	65792	2.072
15	7	2	17	7	2	+1 prefetcher	LRU	MIP.	64768	2.075
15	7	2	17	7	3	+1 prefetcher	LRU	LIP.	65792	2.077
15	7	2	17	7	4	+1 prefetcher	LRU	LIP.	66816	2.080
15	7	2	17	7	3	+1 prefetcher	LFU	MIP.	65792	2.081
15	7	2	17	7	5	+1 prefetcher	LRU	LIP.	67840	2.081
15	7	2	17	7	3	+1 prefetcher	LFU	LIP.	65792	2.082
15	7	2	17	7	4	+1 prefetcher	LFU	MIP.	66816	2.082
15	7	2	17	7	5	+1 prefetcher	LFU	MIP.	67840	2.083
15	7	2	17	7	2	+1 prefetcher	LRU	LIP.	64768	2.084
15	7	2	17	7	5	+1 prefetcher	LFU	LIP.	67840	2.084
15	7	2	17	7	4	+1 prefetcher	LFU	LIP.	66816	2.085
15	7	2	17	7	2	+1 prefetcher	LFU	MIP.	64768	2.096
15	7	2	17	7	2	+1 prefetcher	LFU	LIP.	64768	2.098
15	7	2	17	7	5	Strided prefetcher	LRU	MIP.	67840	2.104
15	7	2	17	7	3	Strided prefetcher	LRU	LIP.	65792	2.105
15	7	2	17	7	4	Strided prefetcher	LRU	MIP.	66816	2.105
15	7	2	17	7	4	Strided prefetcher	LRU	LIP.	66816	2.105
15	7	2	17	7	5	Strided prefetcher	LRU	LIP.	67840	2.105
15	7	2	17	7	3	Strided prefetcher	LRU	MIP.	65792	2.106
15	7	2	17	7	4	Strided prefetcher	LFU	MIP.	66816	2.107
15	7	2	17	7	5	Strided prefetcher	LFU	MIP.	67840	2.107
15	7	2	17	7	2	Strided prefetcher	LRU	LIP.	64768	2.108
15	7	2	17	7	4	Strided prefetcher	LFU	LIP.	66816	2.108
15	7	2	17	7	5	Strided prefetcher	LFU	LIP.	67840	2.108
15	7	2	17	7	2	Strided prefetcher	LRU	MIP.	64768	2.109
15	7	2	17	7	3	Strided prefetcher	LFU	MIP.	65792	2.109
15	7	2	17	7	3	Strided prefetcher	LFU	LIP.	65792	2.109

The selected configuration is as follows:

C1	B1	S1	C2	B2	S2	Prefetch Policy	Replacement Policy	Insertion Policy	Meta_cost	L1_AAT
15	7	2	17	7	5	+1 prefetcher	LRU	MIP	67840	2.070

Appendix:

The python code to anaylsis the output file and generate table:

```python
def parse_cache_settings(file_path):
    with open(file_path, 'r') as file:
        lines = file.readlines()
    return [l.rstrip("\n") for l in lines if len(l) >= 5]

def parse_string(data_str):
    parsed_values = {
        'c1': None,
        'b1': None,
        's1': None,
        'c2': None,
        'b2': None,
        's2': None,
        'prefetch': None,
        'rp': None,
        'IP': None,
    }
    sections = data_str.split('. ')
    l1_values = sections[0].split('(')[1].split(')')[0].split(',')
    parsed_values['c1'] = int(l1_values[0])
    parsed_values['b1'] = int(l1_values[1])
    parsed_values['s1'] = int(l1_values[2])
    parsed_values['rp'] = sections[1].split('.')[0]
    l2_values = sections[3].split('(')[1].split(')')[0].split(',')
    parsed_values['c2'] = int(l2_values[0])
    parsed_values['b2'] = int(l2_values[1])
    parsed_values['s2'] = int(l2_values[2])

    prefetch_value = sections[5]
    parsed_values['prefetch'] = sections[5]

    parsed_values['IP'] = sections[-1]
    return parsed_values

def compute_meta_data(c1,b1,s1,c2,b2,s2):
    meta1 = 2**(c1-b1)*(64-(c1-s1)+2)
    meta2 = 2**(c2-b2)*(64-(c2-s2)+1)
    return meta1 + meta2


from prettytable import PrettyTable

file_dict = {}
file_path = './pb_out/traces/mcf.trace.txt'
file_list = parse_cache_settings(file_path)
for i in range(0, len(file_list), 2):
    num_l = file_list[i+1].split()[0]
```

```
    file_dict[file_list[i]] = num_l

data_list = []
for line in file_dict.keys():
    line_dict = parse_string(line)
    parsed_data = [l for l in line_dict.values()]
    parsed_data.append(compute_meta_data(line_dict['c1'],line_dict['b1'],
line_dict['s1'] \
,line_dict['c2'],line_dict['b2'],line_dict['s2']))
    parsed_data.append(file_dict[line])
    data_list.append(parsed_data)

table = PrettyTable()

sorted_data_lists_by_last_element = sorted(data_list, key=lambda x: x[-1])

column_names = [f"Column {i+1}" for i in range(len(data_list[0]))]
column_names = ["C1","B1", "S1","C2","B2","S2","Prefetch Policy","Replace
Policy","Insertion Policy","Meta_cost","L1_AAT"]
table.field_names = column_names

for one_line in sorted_data_lists_by_last_element[:30]:
    table.add_row(one_line)

print(table)
```

The shell script to generate out for every trace:

```
for script in valid_grad_*.sh; do
    ./"$script" &
done

wait
```