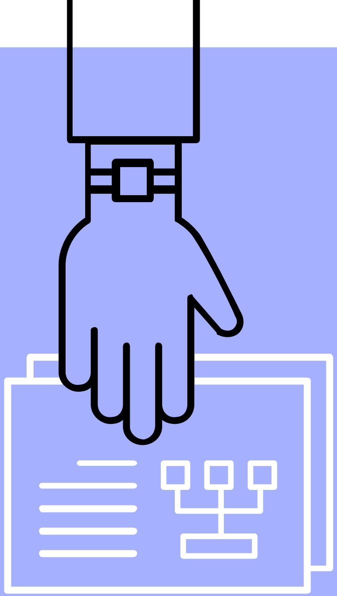
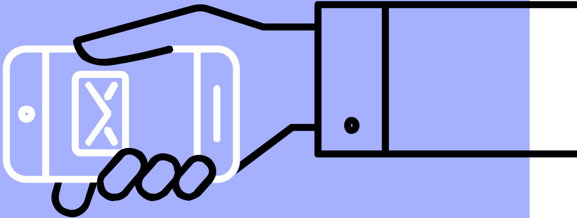
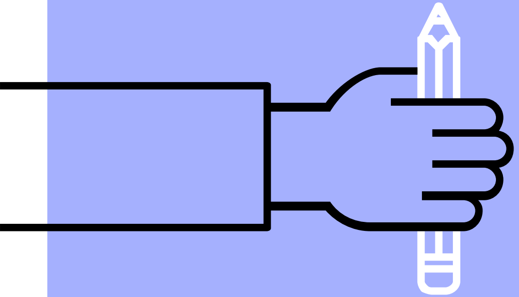
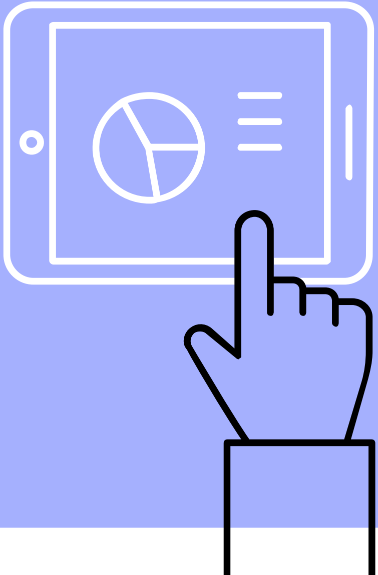




PDede: Partitioned, Deduplicated, Delta Branch Target Buffer





Team: QuickFixDemons

Ashish Kumar Gupta	Sudhir Kumar	Amit Meena	Sourav Kumar	Siddharth Choudhary
180050013	170050053	190050014	180050105	160050036

What is Branch Target Buffer (BTB)?

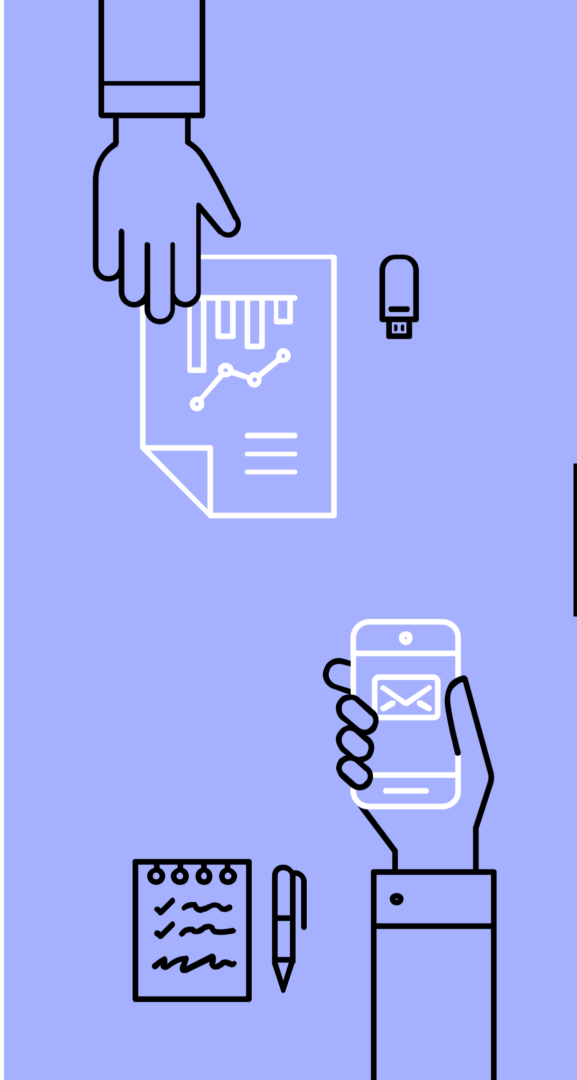
- Maps PC to Target Address
- Helps CPU to find next instruction to execute
- Only Branch Instruction

Address	Target Address
PC	PC + 39

Motivation

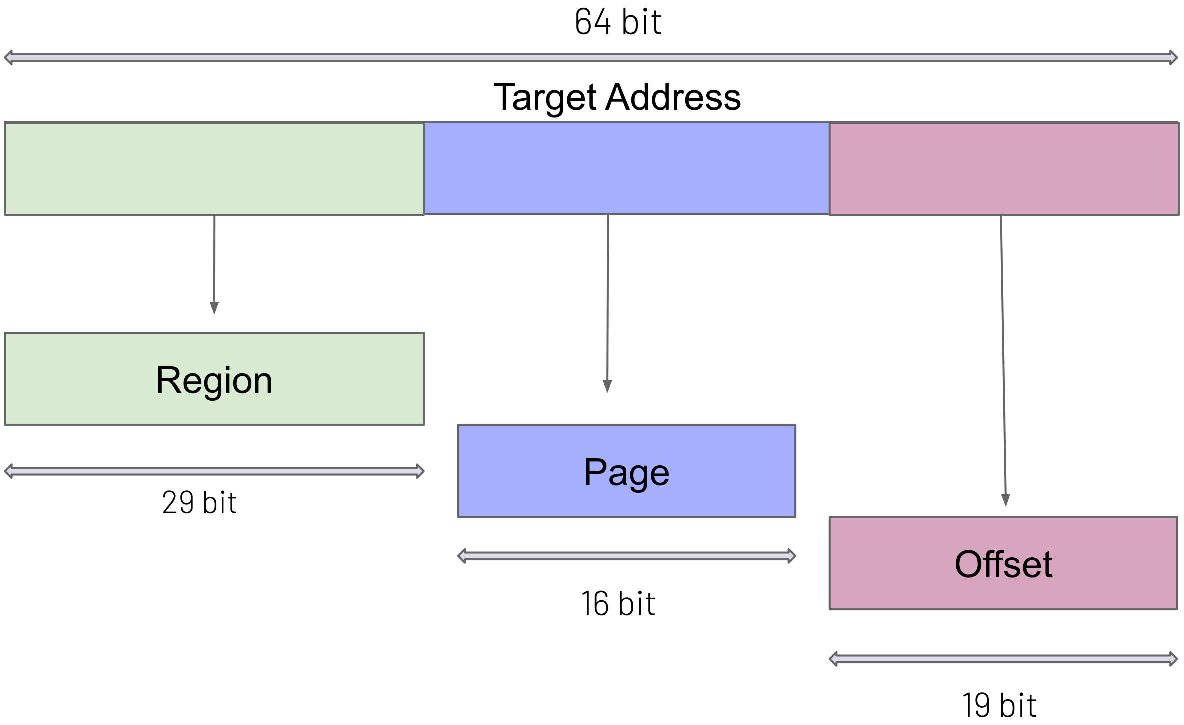
- Large code Footprint
- CPU frontend bottleneck
- Contribution of BTB misses
- Limited BTB capacity

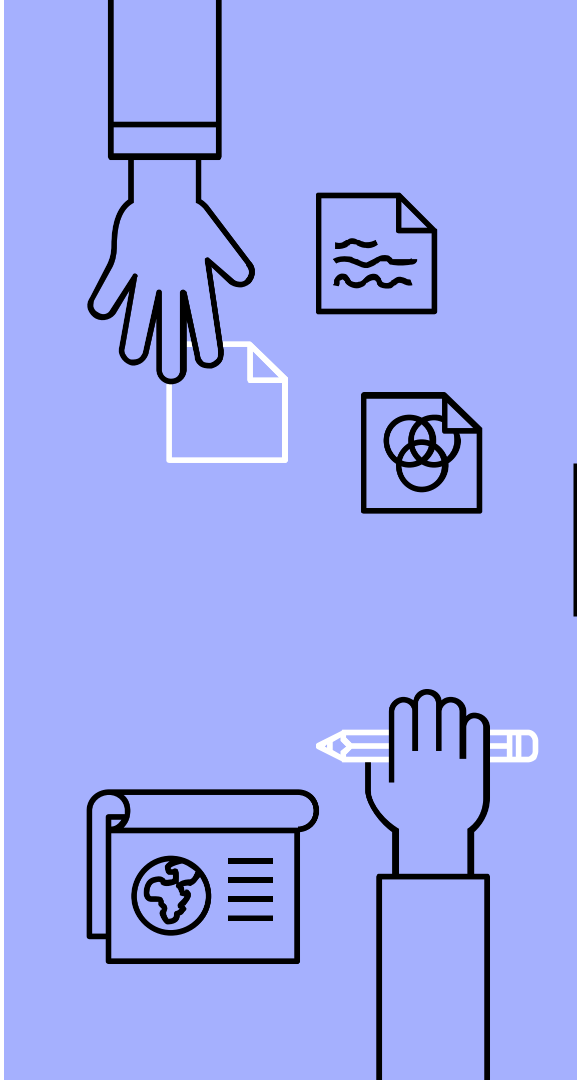
For instance, Google has reported that 23.5% of all CPU cycles are lost to frontend stalls for its Web search binary .



Partitioning

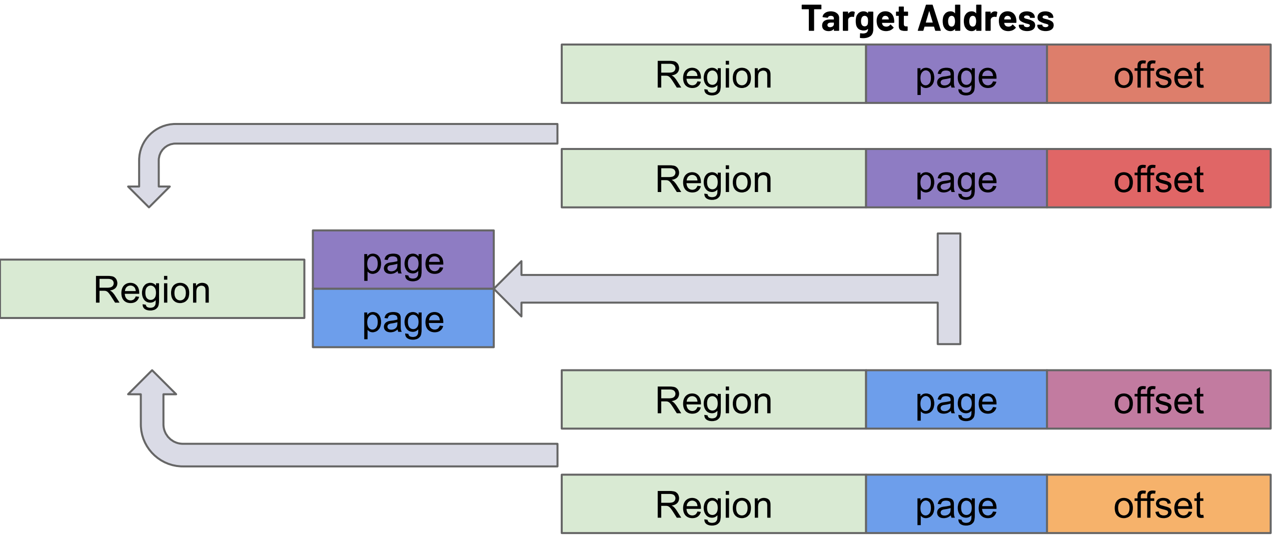
- Target Address can be decomposed
- Improve storage space efficiency

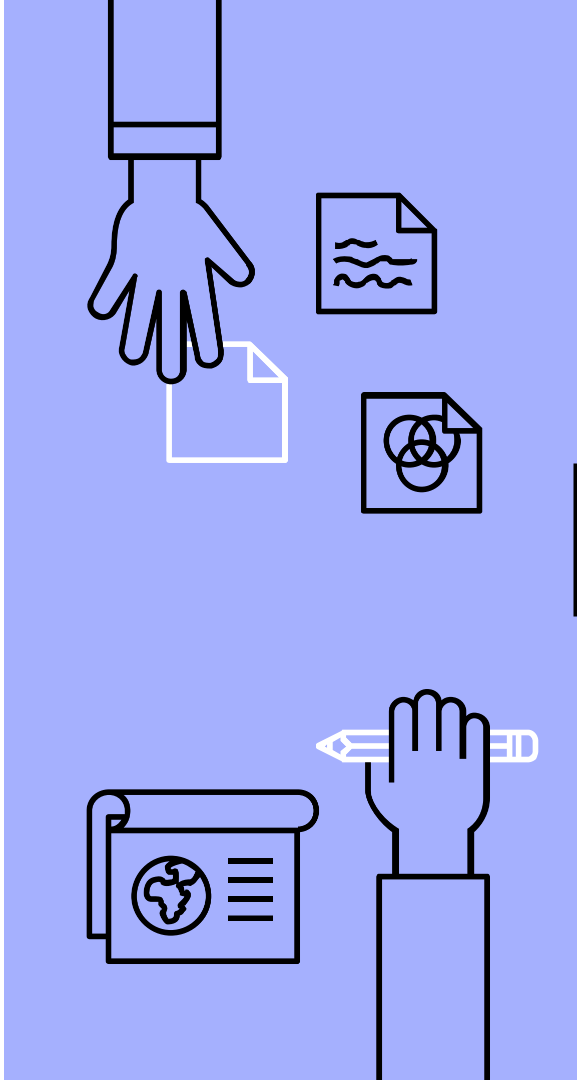




Target Deduplication

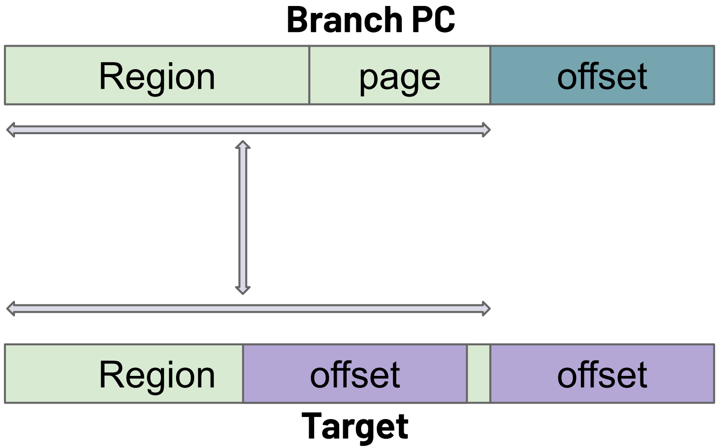
- Many Target Address maps to same region and pages

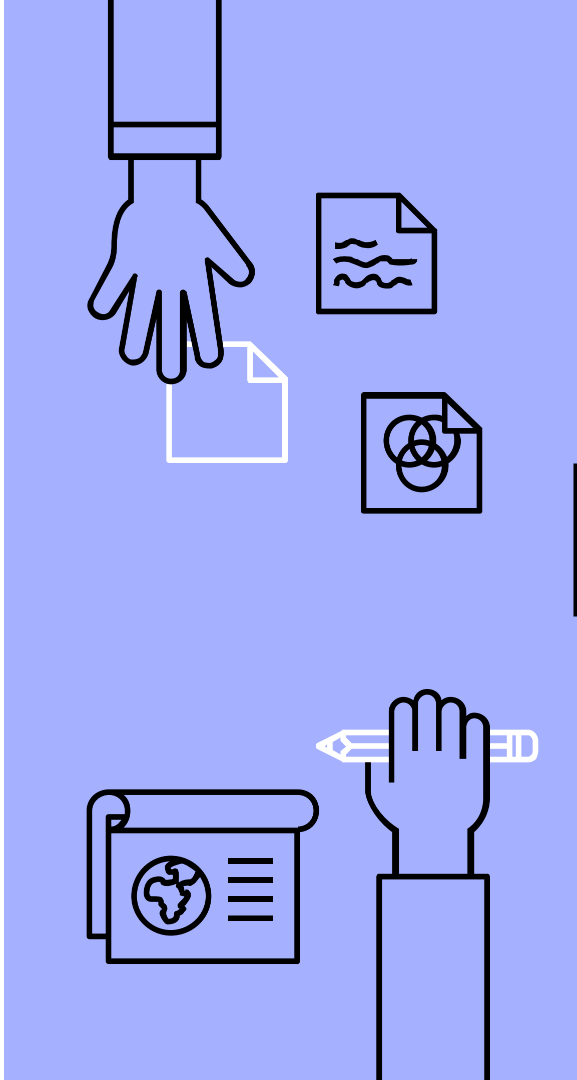




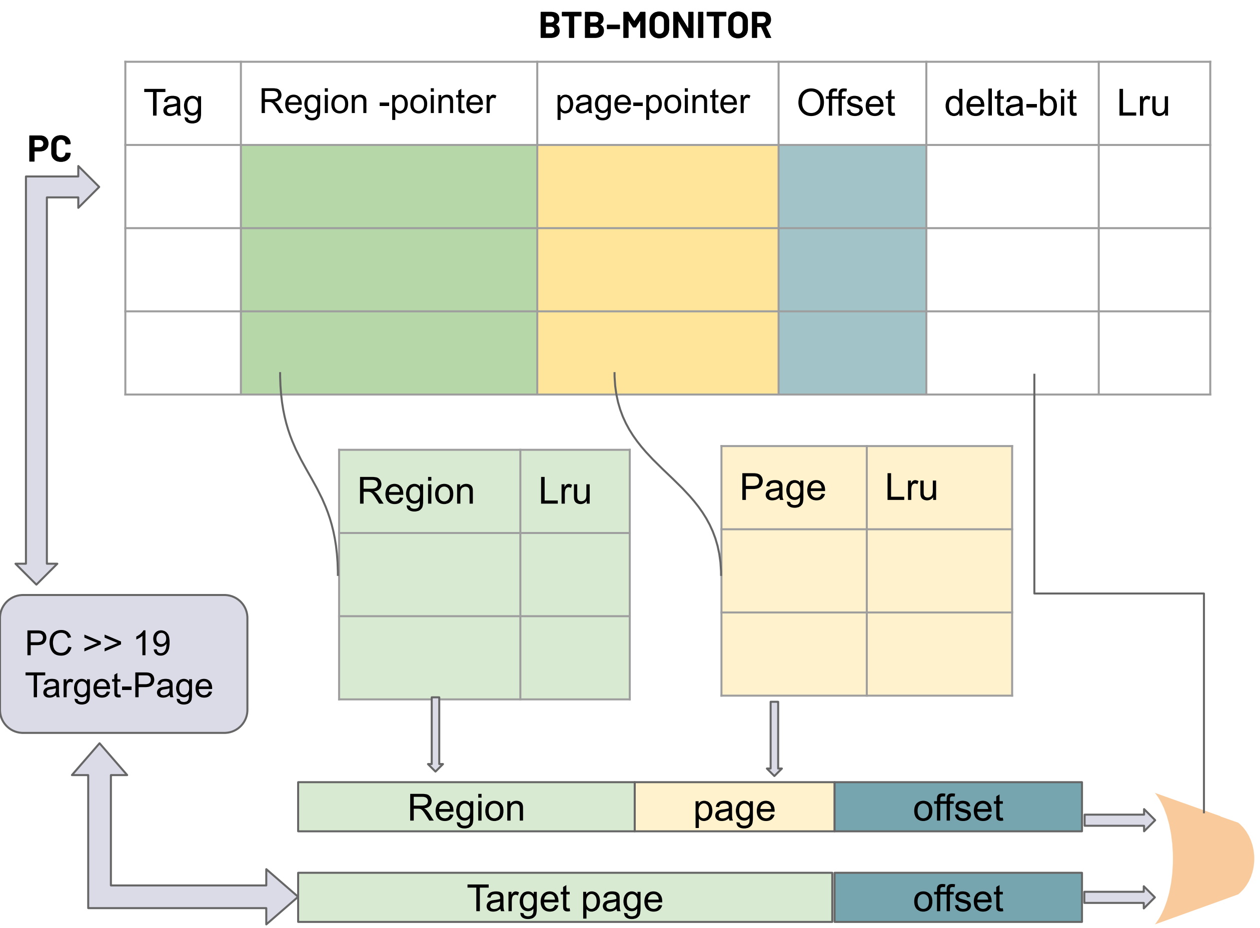
Target Encoding

- Branch PC and Target often share the same region and page





ALL TOGETHER

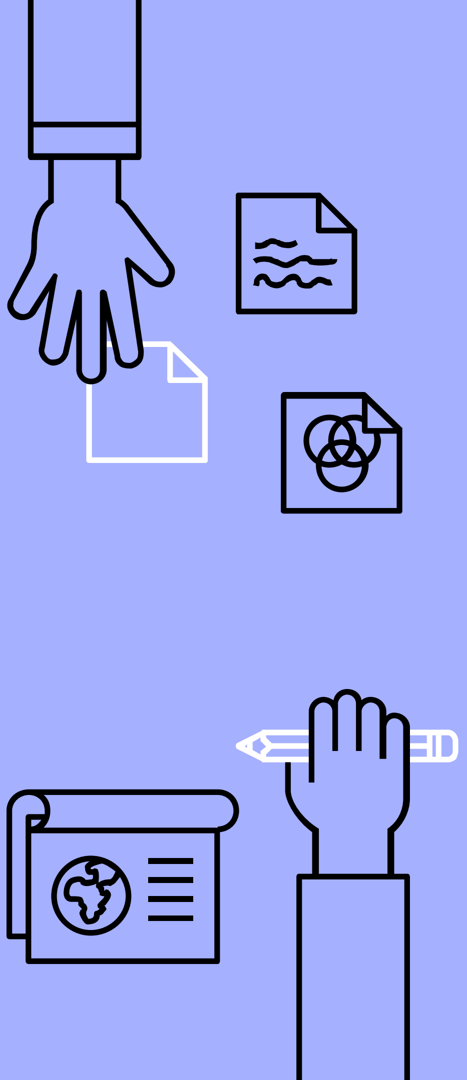


BTB-MONITOR

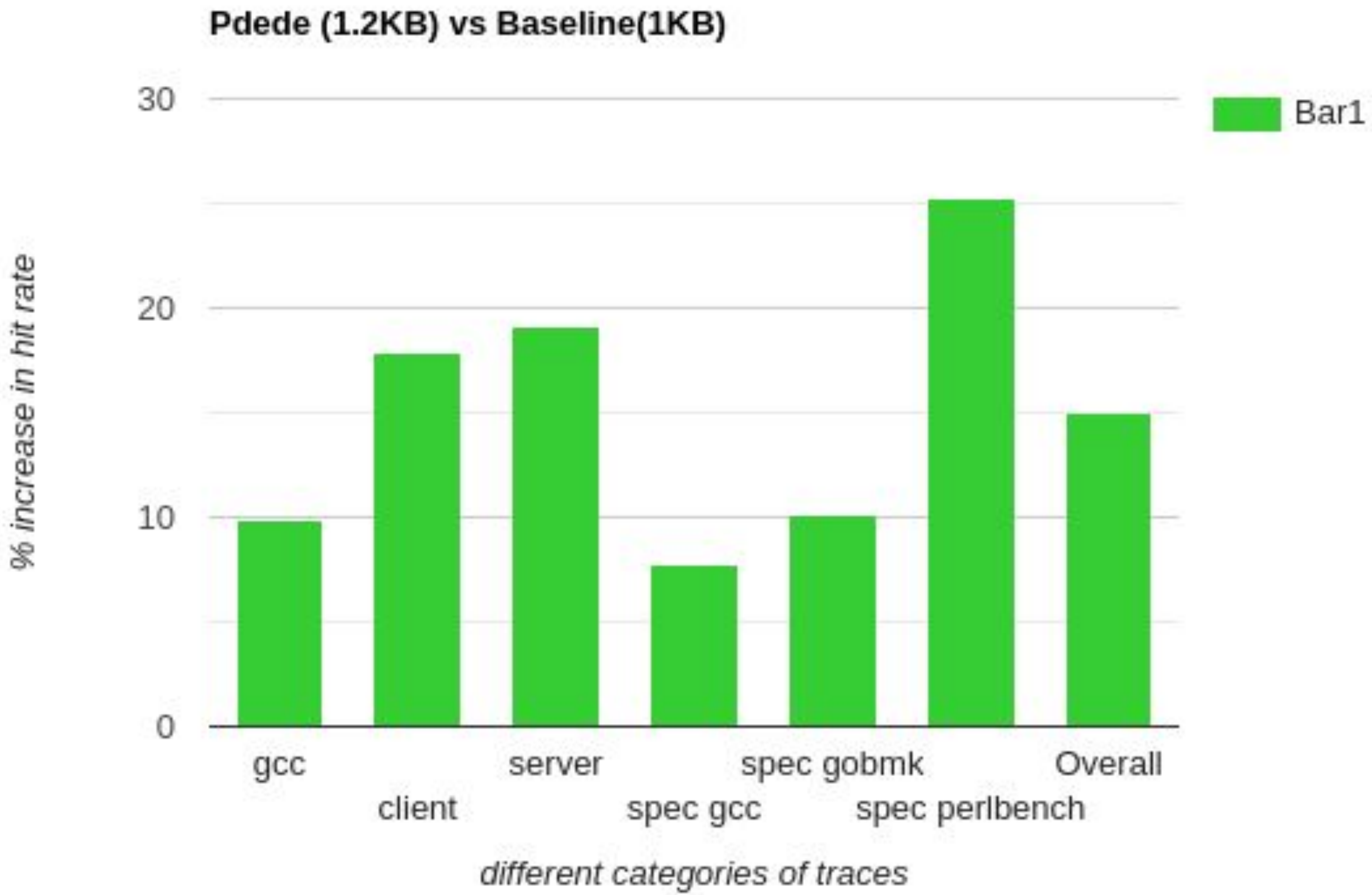
Tag	Region -pointer	page-pointer	Offset	delta-bit	Lru

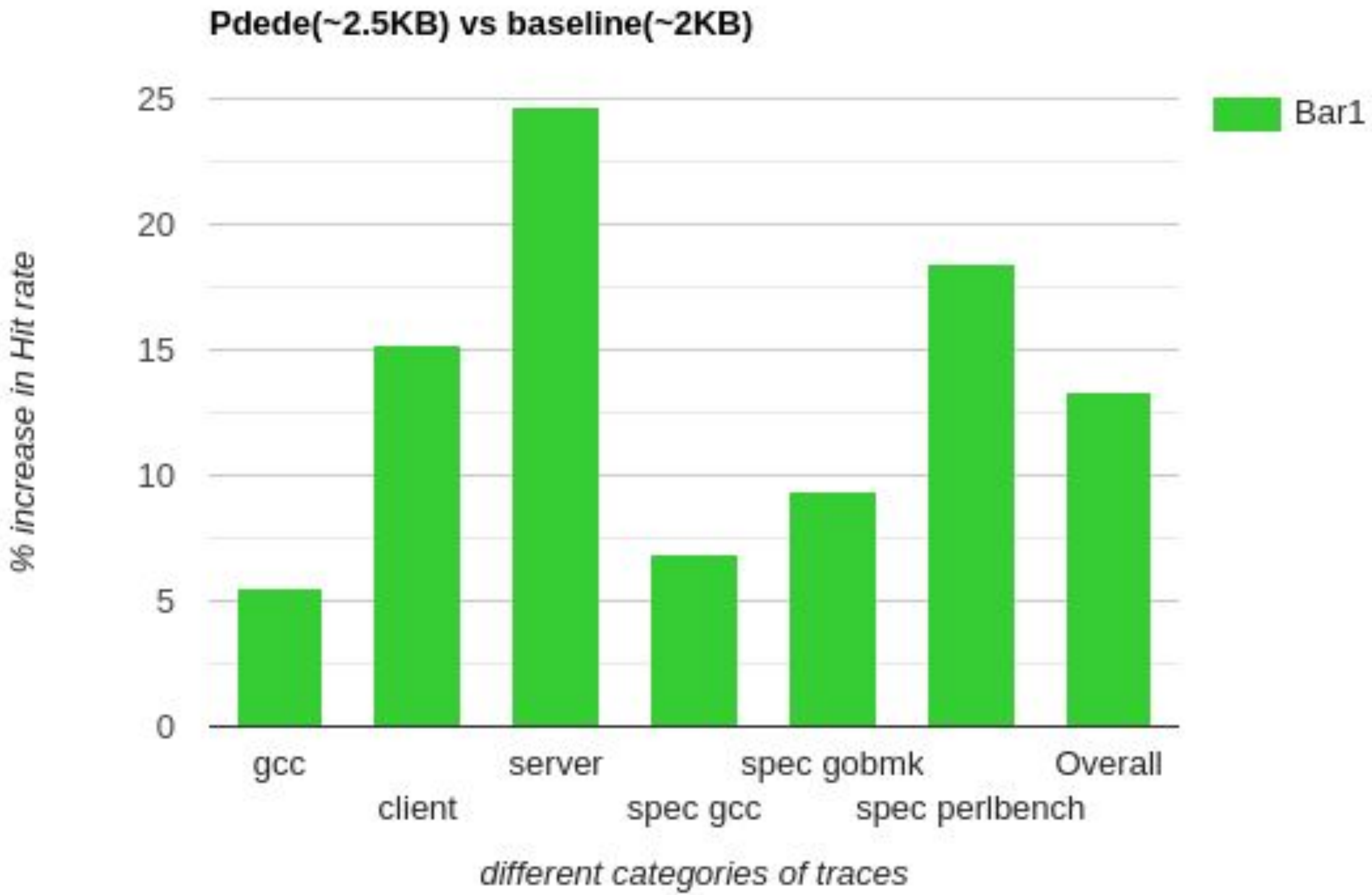
Region	Lru

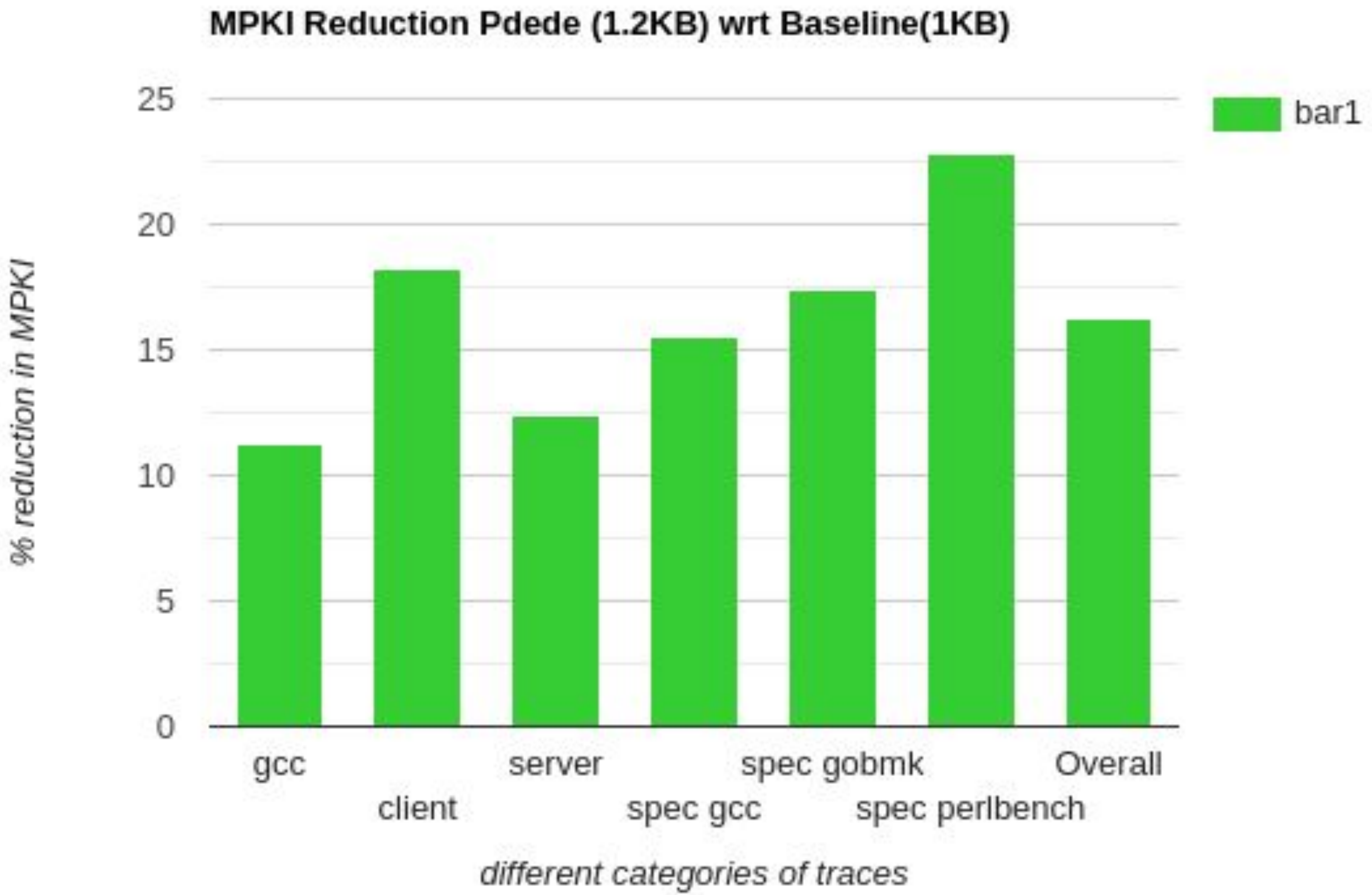
Page	Lru

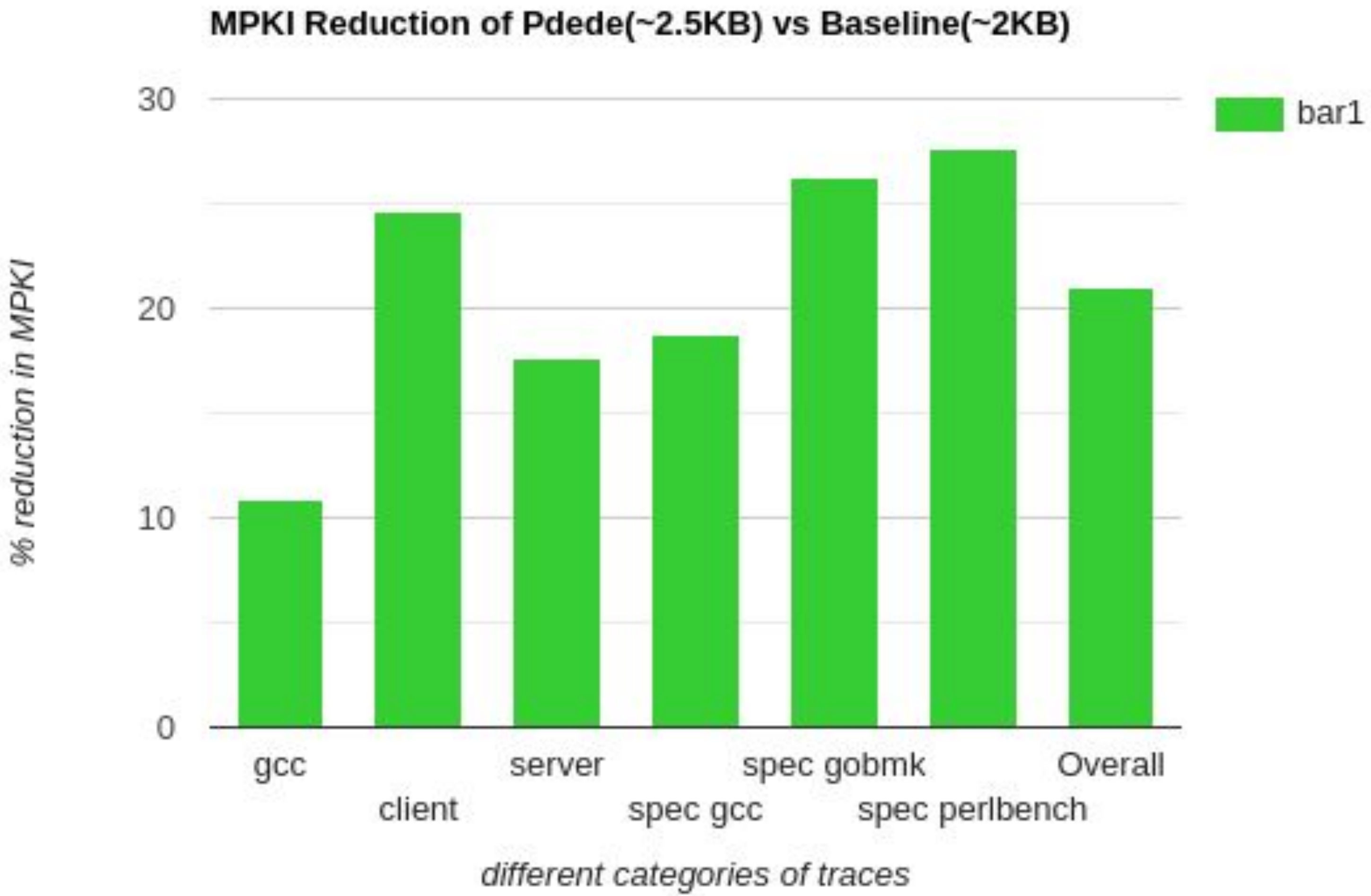


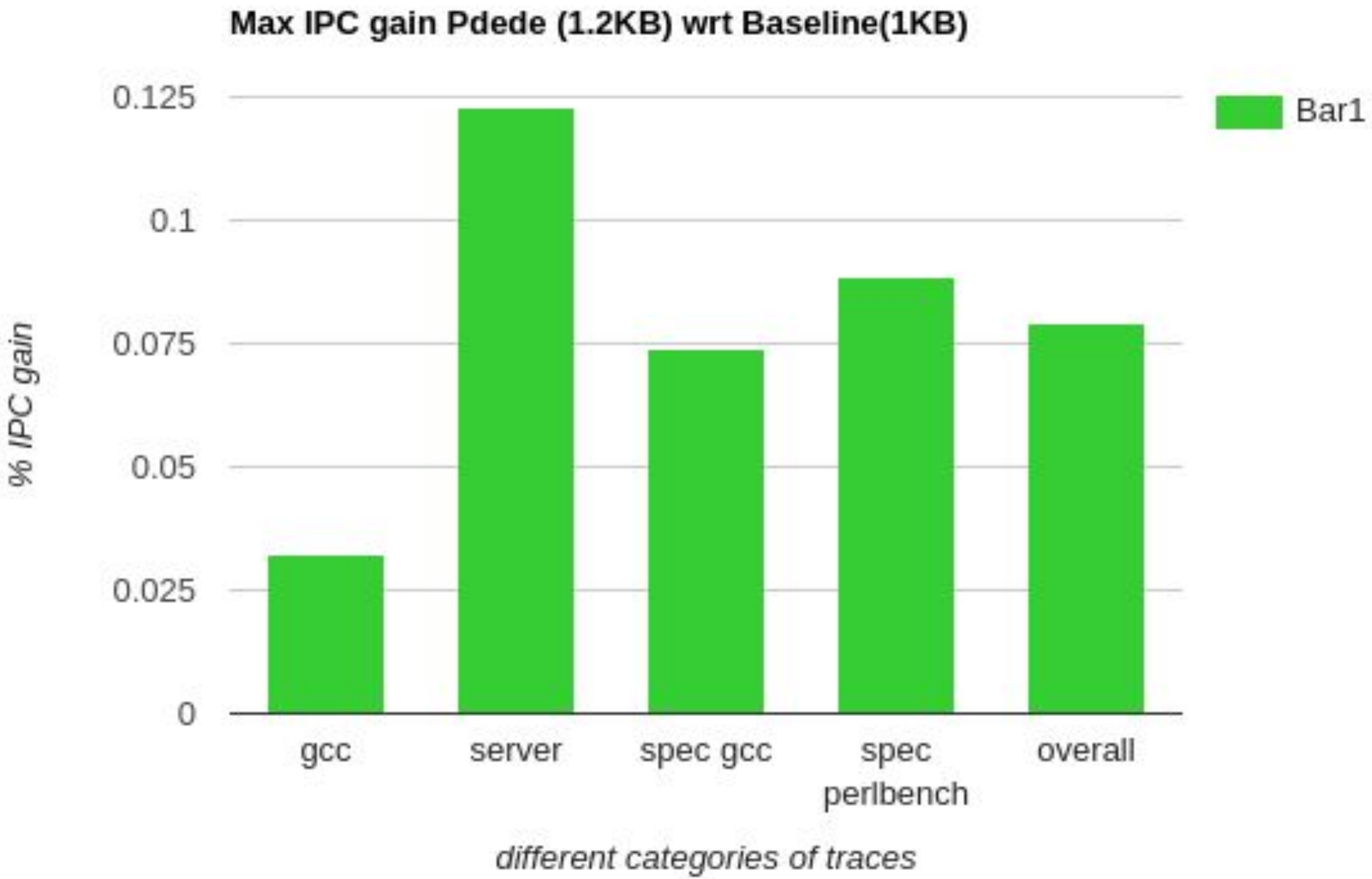
RESULTS


Pdede (1.2KB) vs Baseline(1KB)
🟩 Bar1
% increase in hit rate
30
20
10
0
gcc
client
server
spec gcc
spec gobmk
spec perlbench
Overall
different categories of traces


Pdede(~2.5KB) vs baseline(~2KB)
25
20
15
10
5
0
% increase in Hit rate
🟩 Bar1
gcc
client
server
spec gcc
spec gobmk
spec perlbench
Overall
different categories of traces


MPKI Reduction Pdede (1.2KB) wrt Baseline(1KB)
% reduction in MPKI
25
20
15
10
5
0
🟩 bar1
gcc
client
server
spec gcc
spec gobmk
spec perlbench
Overall
different categories of traces


MPKI Reduction of Pdede(~2.5KB) vs Baseline(~2KB)
30
20
10
0
% reduction in MPKI
🟩 bar1
gcc
client
server
spec gcc
spec gobmk
spec perlbench
Overall
different categories of traces


Max IPC gain Pdede (1.2KB) wrt Baseline(1KB)
🟩 Bar1
% IPC gain
0.125
0.1
0.075
0.05
0.025
0
gcc
server
spec gcc
spec perlbench
overall
different categories of traces

Pros and Cons

- Space Efficient
- Reduces Miss Rate

Pros

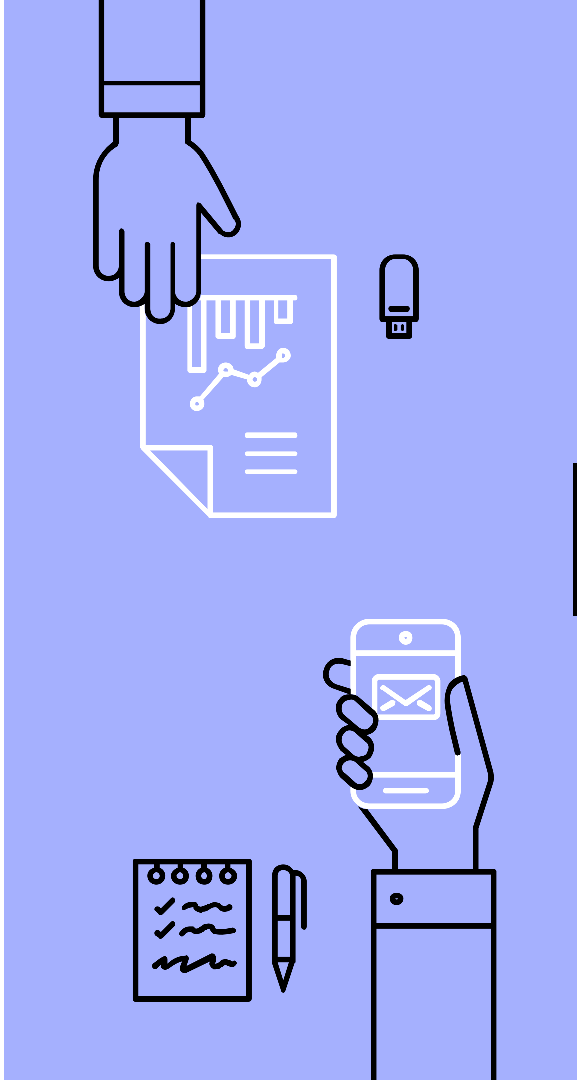
- Works well only on application having high percentage of branch
- Large code-footprint
- Data-center applications

Cons

Conclusion

- Three ideas
- Space-Efficiency of BTBs
- Performance Gains

The novel ideas presented and these insights will ease future research on increasing efficiency and throughput of branch target buffers.



THANKS!

Any questions?

You can find us at:

180050013@iitb.ac.in

Repository:

BTB Organisation