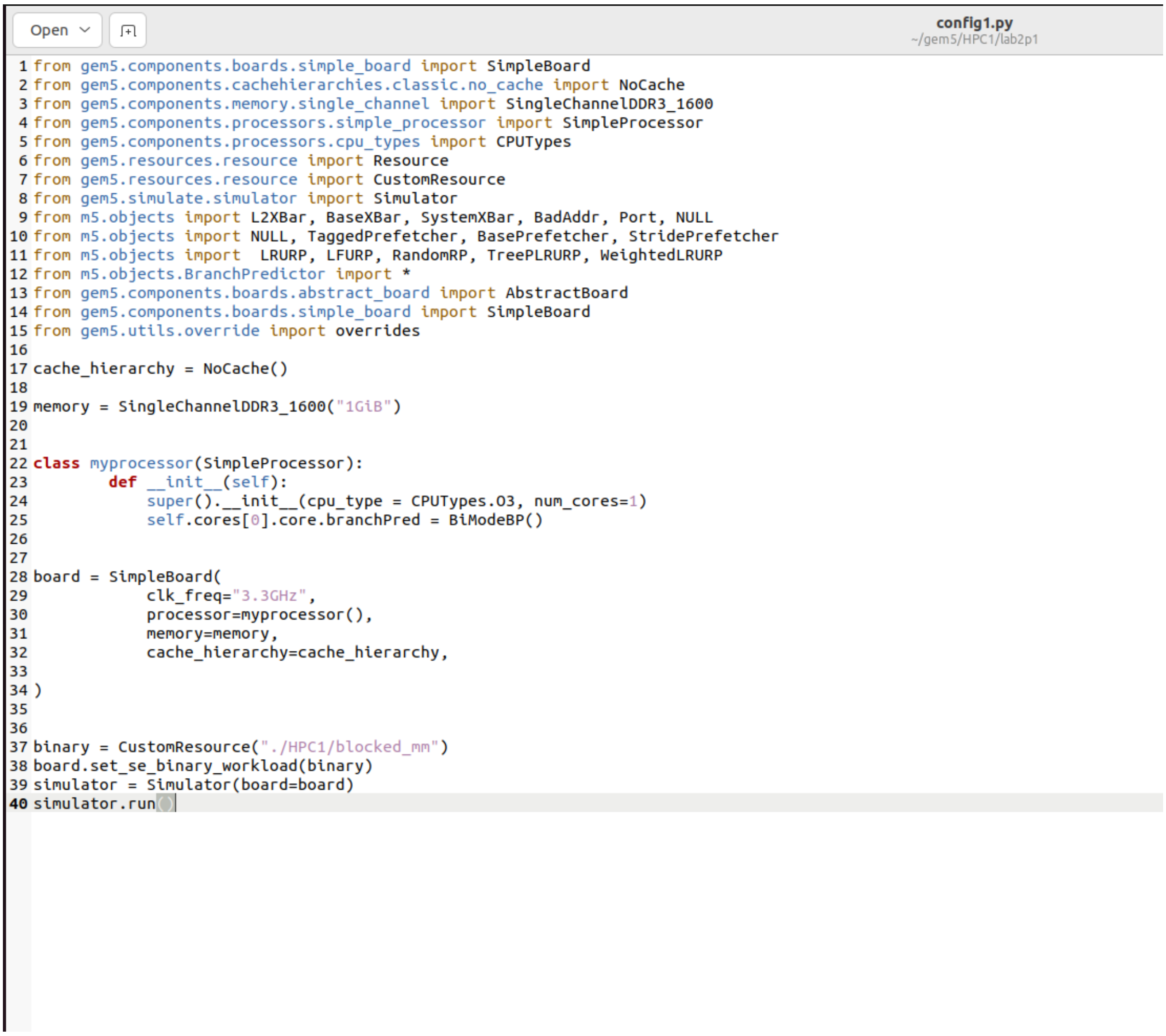
Lab Assignment-2

By Tanmay Aron (2023JVL2240)

Branch Prediction

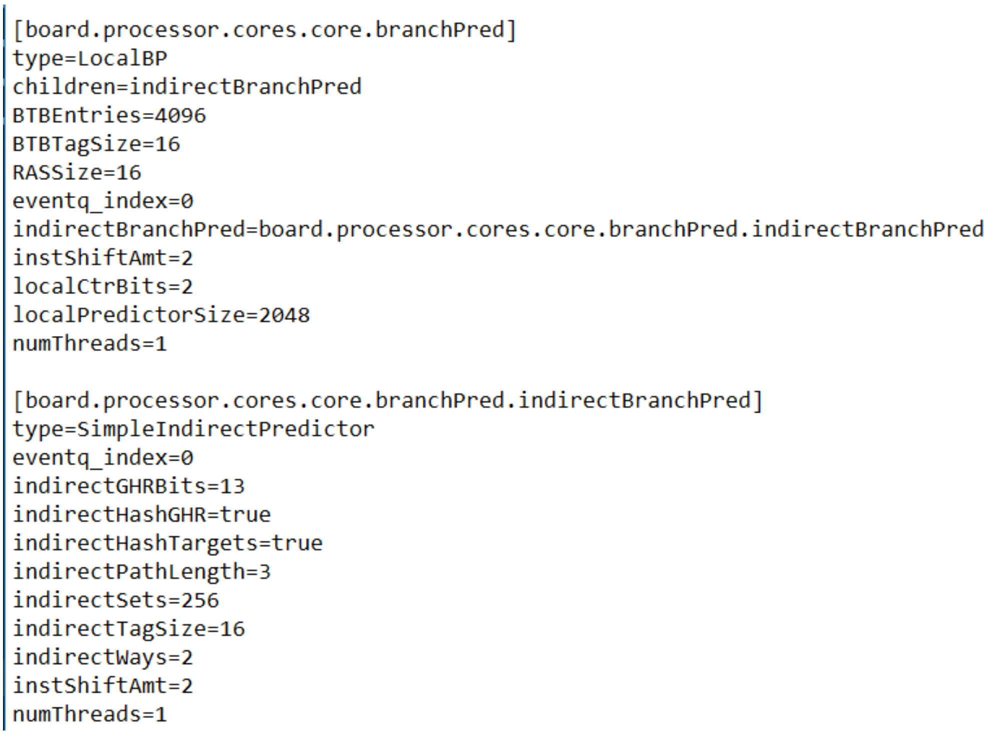
Config file:-



```python
from gem5.components.boards.simple_board import SimpleBoard
from gem5.components.cachehierarchies.classic.no_cache import NoCache
from gem5.components.memory.single_channel import SingleChannelDDR3_1600
from gem5.components.processors.simple_processor import SimpleProcessor
from gem5.components.processors.cpu_types import CPUTypes
from gem5.resources.resource import Resource
from gem5.resources.resource import CustomResource
from gem5.simulate.simulator import Simulator
from m5.objects import L2XBar, BaseXBar, SystemXBar, BadAddr, Port, NULL
from m5.objects import NULL, TaggedPrefetcher, BasePrefetcher, StridePrefetcher
from m5.objects import  LRURP, LFURP, RandomRP, TreePLRURP, WeightedLRURP
from m5.objects.BranchPredictor import *
from gem5.components.boards.abstract_board import AbstractBoard
from gem5.components.boards.simple_board import SimpleBoard
from gem5.utils.override import overrides

cache_hierarchy = NoCache()

memory = SingleChannelDDR3_1600("1GiB")


class myprocessor(SimpleProcessor):
        def __init__(self):
            super().__init__(cpu_type = CPUTypes.O3, num_cores=1)
            self.cores[0].core.branchPred = BiModeBP()


board = SimpleBoard(
            clk_freq="3.3GHz",
            processor=myprocessor(),
            memory=memory,
            cache_hierarchy=cache_hierarchy,

)


binary = CustomResource("./HPC1/blocked_mm")
board.set_se_binary_workload(binary)
simulator = Simulator(board=board)
simulator.run()
```

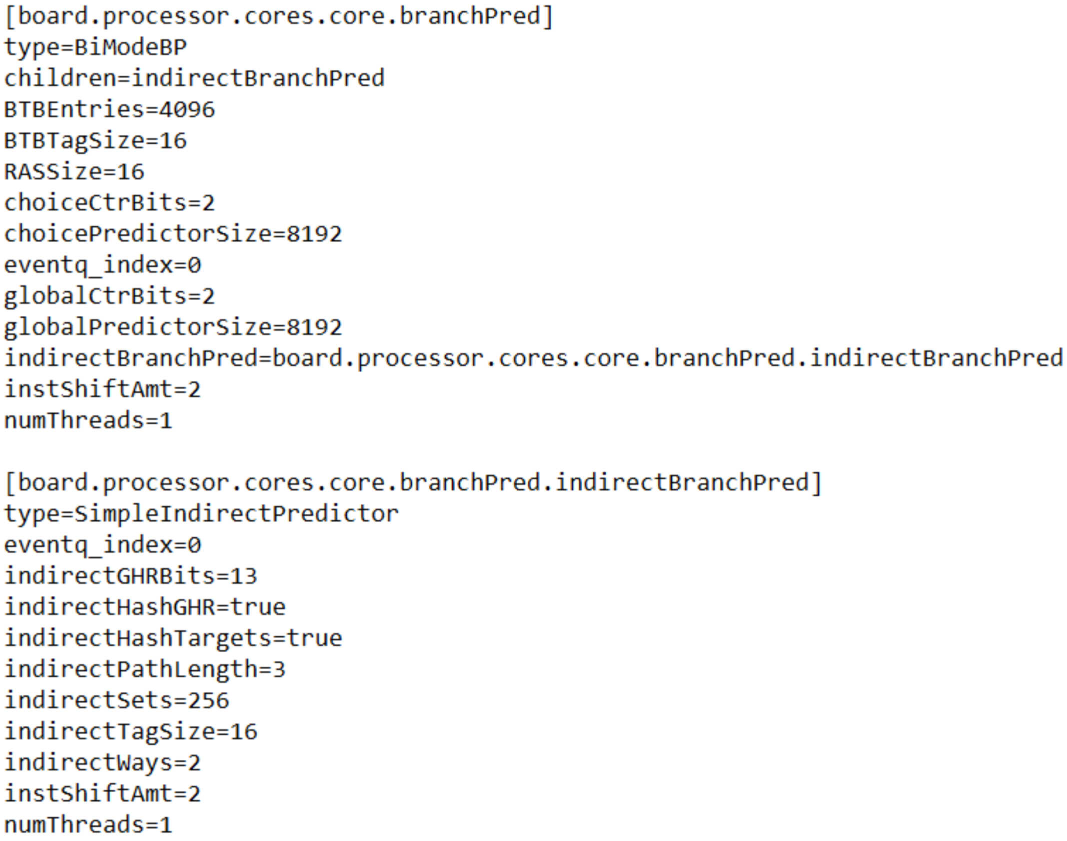
Created a class myprocessor() which allowed us to add branch predictor. The three predictors we used are: Tournament Branch Predictor, Local Branch predictor and BiMode Branch Predictor.

Config.ini file for Local Branch Predictor



```
[board.processor.cores.core.branchPred]
type=LocalBP
children=indirectBranchPred
BTBEntries=4096
BTBTagSize=16
RASSize=16
eventq_index=0
indirectBranchPred=board.processor.cores.core.branchPred.indirectBranchPred
instShiftAmt=2
localCtrBits=2
localPredictorSize=2048
numThreads=1

[board.processor.cores.core.branchPred.indirectBranchPred]
type=SimpleIndirectPredictor
eventq_index=0
indirectGHRBits=13
indirectHashGHR=true
indirectHashTargets=true
indirectPathLength=3
indirectSets=256
indirectTagSize=16
indirectWays=2
instShiftAmt=2
numThreads=1
```

Config.ini file for BiMode Branch Predictor

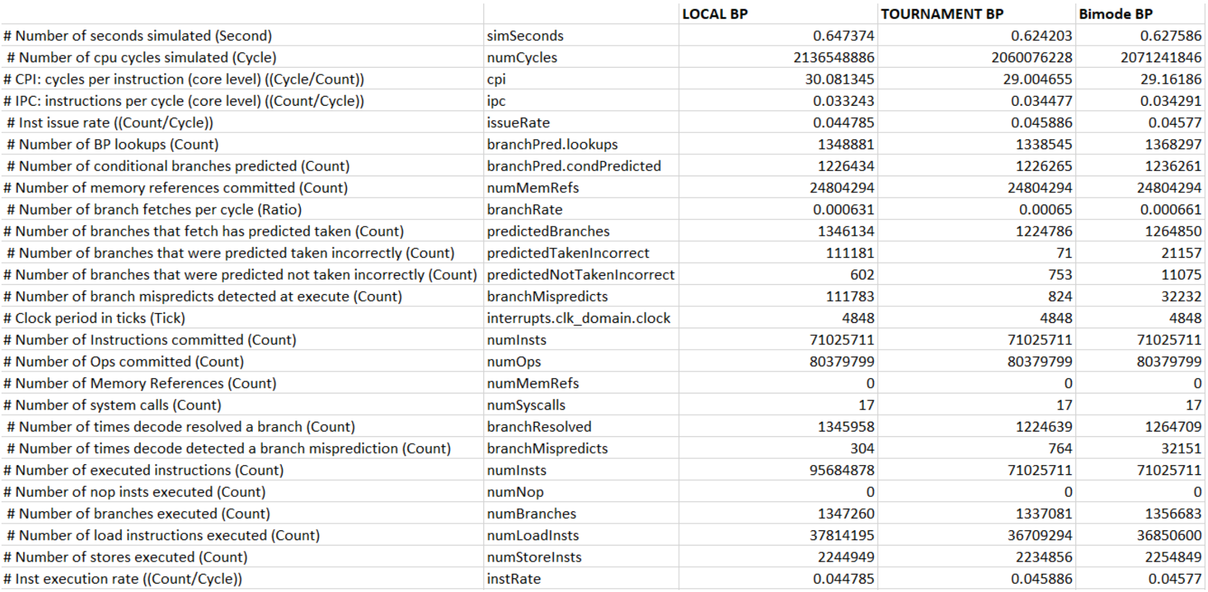


```
[board.processor.cores.core.branchPred]
type=BiModeBP
children=indirectBranchPred
BTBEntries=4096
BTBTagSize=16
RASSize=16
choiceCtrBits=2
choicePredictorSize=8192
eventq_index=0
globalCtrBits=2
globalPredictorSize=8192
indirectBranchPred=board.processor.cores.core.branchPred.indirectBranchPred
instShiftAmt=2
numThreads=1

[board.processor.cores.core.branchPred.indirectBranchPred]
type=SimpleIndirectPredictor
eventq_index=0
indirectGHRBits=13
indirectHashGHR=true
indirectHashTargets=true
indirectPathLength=3
indirectSets=256
indirectTagSize=16
indirectWays=2
instShiftAmt=2
numThreads=1
```

Config.ini file for Tournament branch predictor

```
[board.processor.cores.core.branchPred]
type=TournamentBP
children=indirectBranchPred
BTBEntries=4096
BTBTagSize=16
RASSize=16
choiceCtrBits=2
choicePredictorSize=8192
eventq_index=0
globalCtrBits=2
globalPredictorSize=8192
indirectBranchPred=board.processor.cores.core.branchPred.indirectBranchPred
instShiftAmt=2
localCtrBits=2
localHistoryTableSize=2048
localPredictorSize=2048
numThreads=1

[board.processor.cores.core.branchPred.indirectBranchPred]
type=SimpleIndirectPredictor
eventq_index=0
indirectGHRBits=13
indirectHashGHR=true
indirectHashTargets=true
indirectPathLength=3
indirectSets=256
indirectTagSize=16
indirectWays=2
instShiftAmt=2
numThreads=1
```

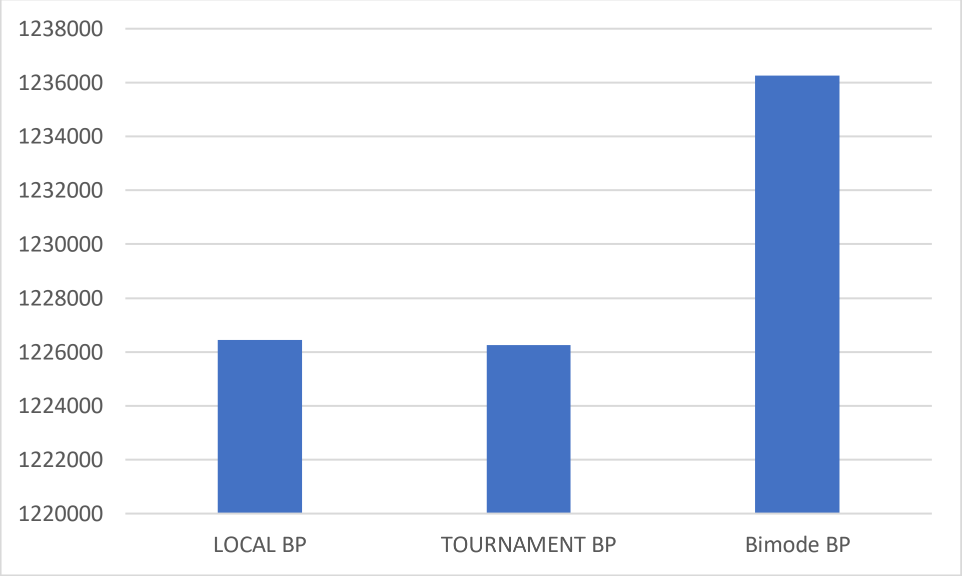
Collected Data



		LOCAL BP	TOURNAMENT BP	Bimode BP
# Number of seconds simulated (Second)	simSeconds	0.647374	0.624203	0.627586
# Number of cpu cycles simulated (Cycle)	numCycles	2136548886	2060076228	2071241846
# CPI: cycles per instruction (core level) ((Cycle/Count))	cpi	30.081345	29.004655	29.16186
# IPC: instructions per cycle (core level) ((Count/Cycle))	ipc	0.033243	0.034477	0.034291
# Inst issue rate ((Count/Cycle))	issueRate	0.044785	0.045886	0.04577
# Number of BP lookups (Count)	branchPred.lookups	1348881	1338545	1368297
# Number of conditional branches predicted (Count)	branchPred.condPredicted	1226434	1226265	1236261
# Number of memory references committed (Count)	numMemRefs	24804294	24804294	24804294
# Number of branch fetches per cycle (Ratio)	branchRate	0.000631	0.00065	0.000661
# Number of branches that fetch has predicted taken (Count)	predictedBranches	1346134	1224786	1264850
# Number of branches that were predicted taken incorrectly (Count)	predictedTakenIncorrect	111181	71	21157
# Number of branches that were predicted not taken incorrectly (Count)	predictedNotTakenIncorrect	602	753	11075
# Number of branch mispredicts detected at execute (Count)	branchMispredicts	111783	824	32232
# Clock period in ticks (Tick)	interrupts.clk_domain.clock	4848	4848	4848
# Number of Instructions committed (Count)	numInsts	71025711	71025711	71025711
# Number of Ops committed (Count)	numOps	80379799	80379799	80379799
# Number of Memory References (Count)	numMemRefs	0	0	0
# Number of system calls (Count)	numSyscalls	17	17	17
# Number of times decode resolved a branch (Count)	branchResolved	1345958	1224639	1264709
# Number of times decode detected a branch misprediction (Count)	branchMispredicts	304	764	32151
# Number of executed instructions (Count)	numInsts	95684878	71025711	71025711
# Number of nop insts executed (Count)	numNop	0	0	0
# Number of branches executed (Count)	numBranches	1347260	1337081	1356683
# Number of load instructions executed (Count)	numLoadInsts	37814195	36709294	36850600
# Number of stores executed (Count)	numStoreInsts	2244949	2234856	2254849
# Inst execution rate ((Count/Cycle))	instRate	0.044785	0.045886	0.04577

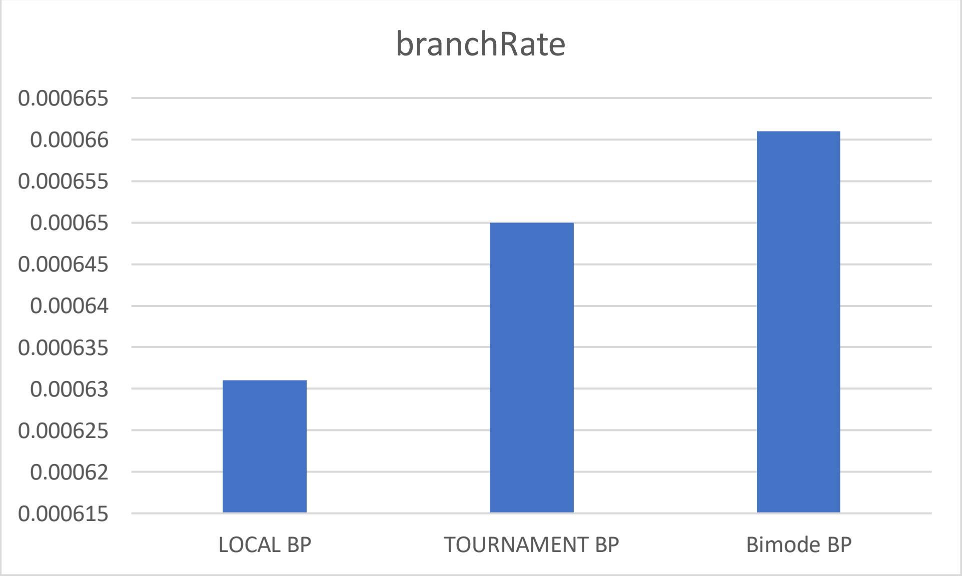
We can notice that simulation time for all the branch predictors doesn’t have much difference.

Number of Conditional Branch predicted-



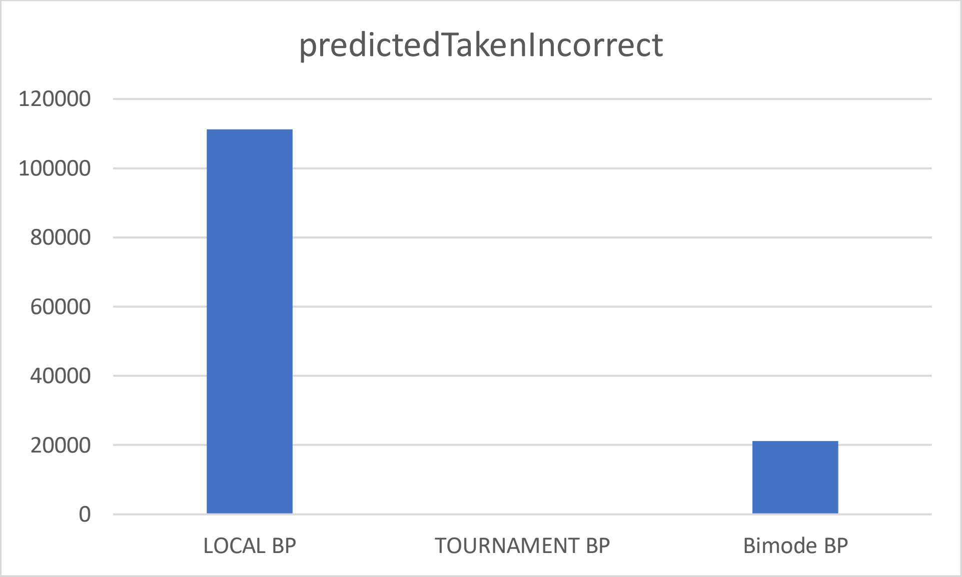
Here, BiMode BP has predicted more number of branches as compared to Local BP and Tournament BP

Number of branch fetches per cycle



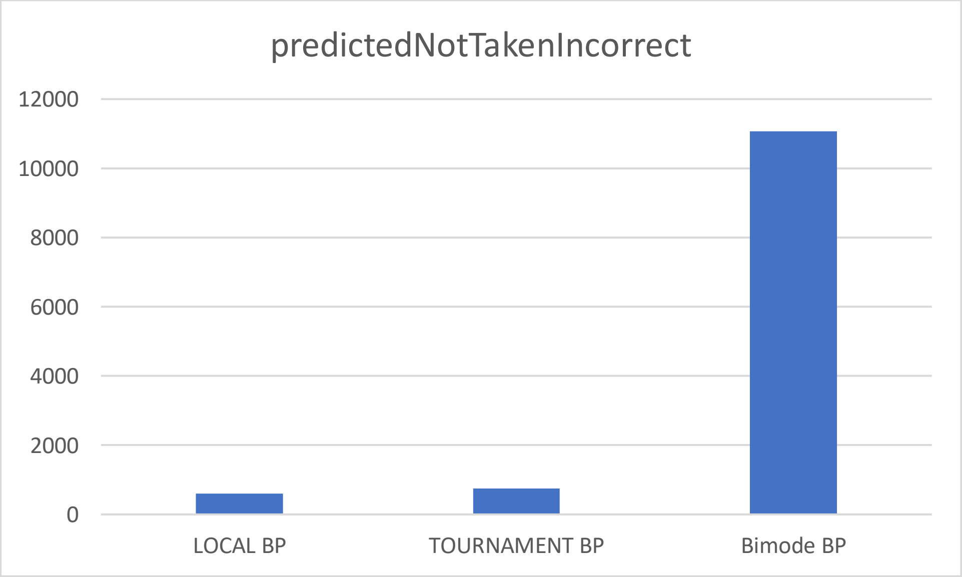
Local BP fetches least number of instructions per cycle and BiMode BP fetches most number of instructions

Number of Predicted Branches taken incorrectly-



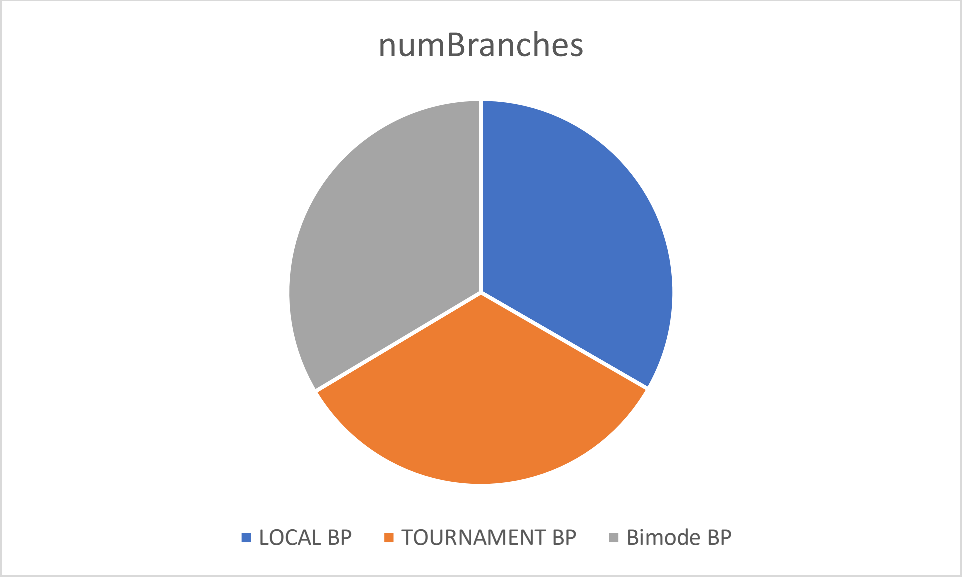
Local BP is giving more incorrect predictions. So far, BiMode BP is giving us great results as number of predictions is more and less mistakes. But, accuracy of Tournament BP is high.

Number of branches that were predicted not taken incorrectly



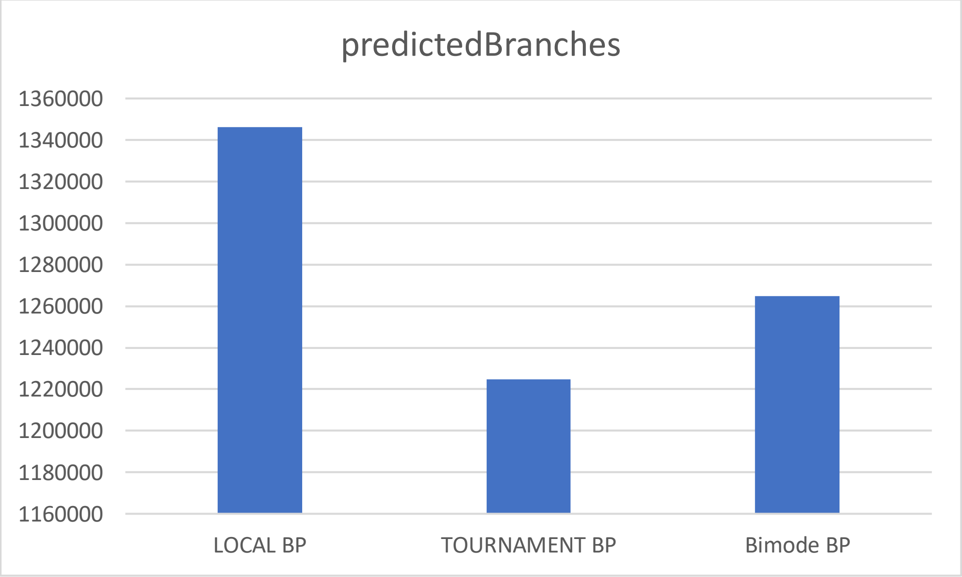
We can see that more number of predictions by BiMode BP isn’t taken which were incorrect.

Number of branches executed

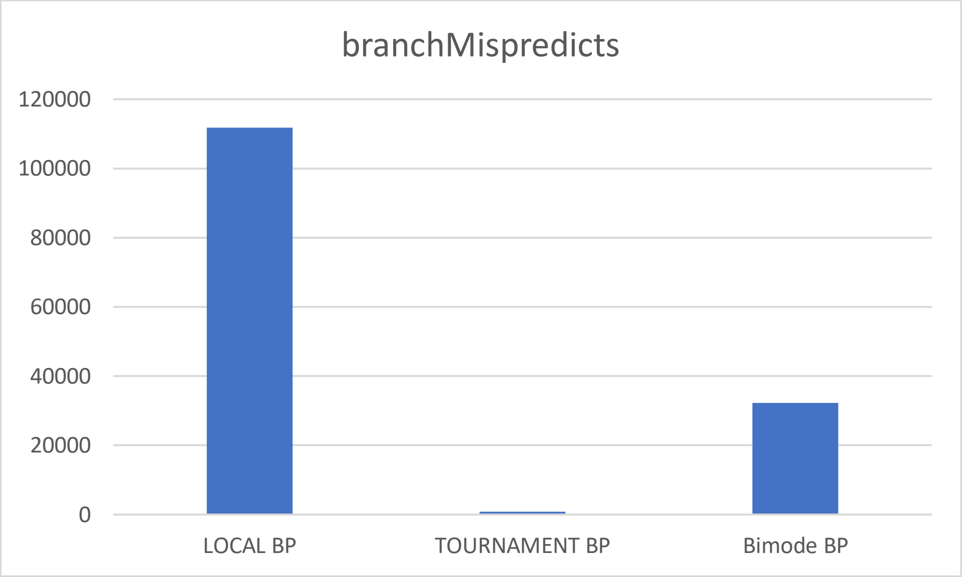


As expected, number of branches executed in all three conditions are similar.

Number of predictions taken



Number of branch mispredicts detected at execute stage



RESULT: We conclude that for large data BiMode BP is the most efficient predictor. It has highest branch fetch rate and gives decent correct predictions.

But, if we want more accurate predictions, we should use Tournament BP as it gives least number of incorrect predictions.