Sampling Methods

ESESC Tutorial

Speaker: Sina Hassani

Department of Computer Engineering,
University of California, Santa Cruz
http://masc.soe.ucsc.edu

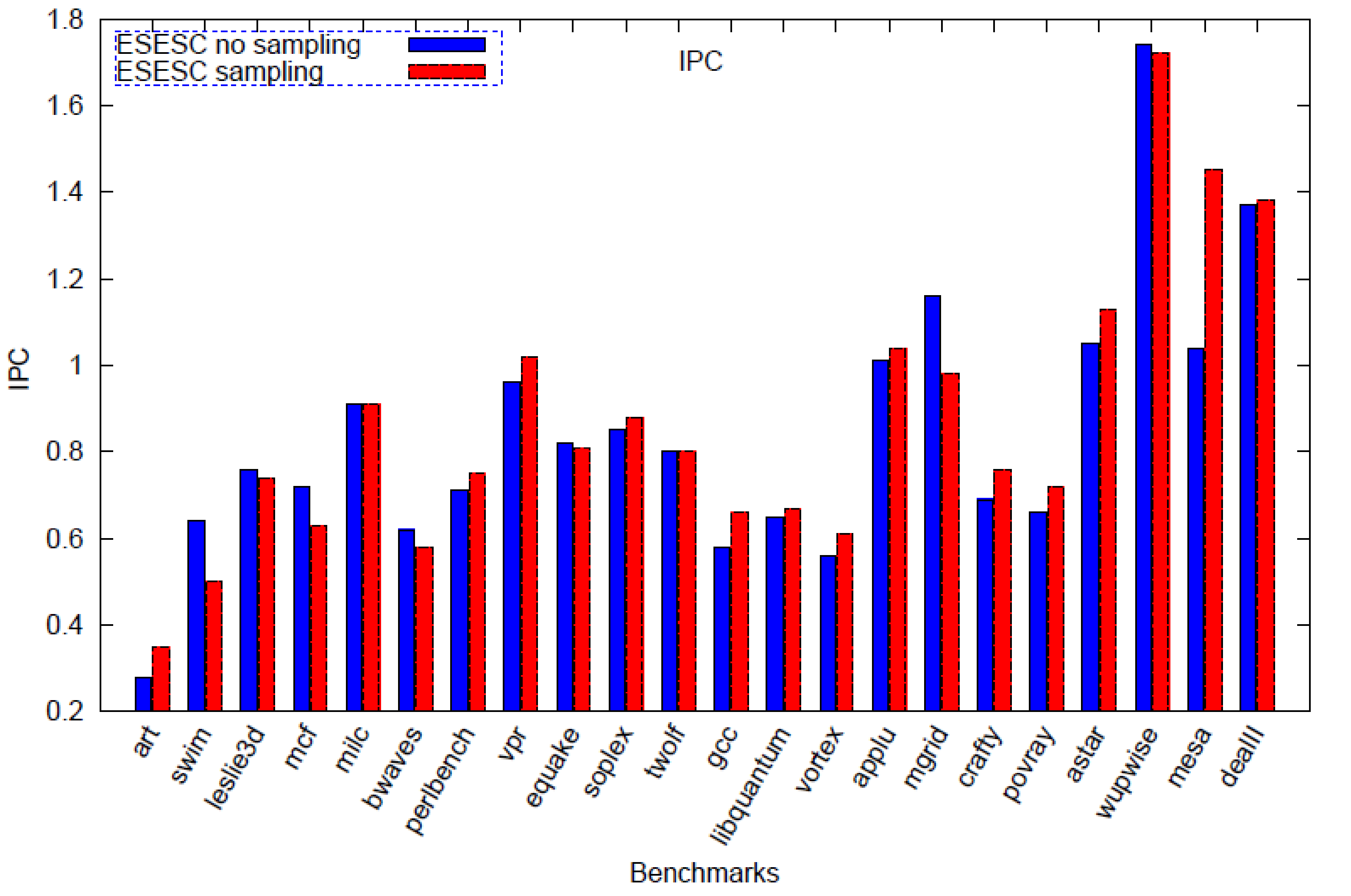
Sampling Methods

- You will learn:
  - Why we need sampling in simulation
  - How time-based sampling works
  - Sampling configurations in ESESC
  - How to use Gstats to add add a new statistics counter

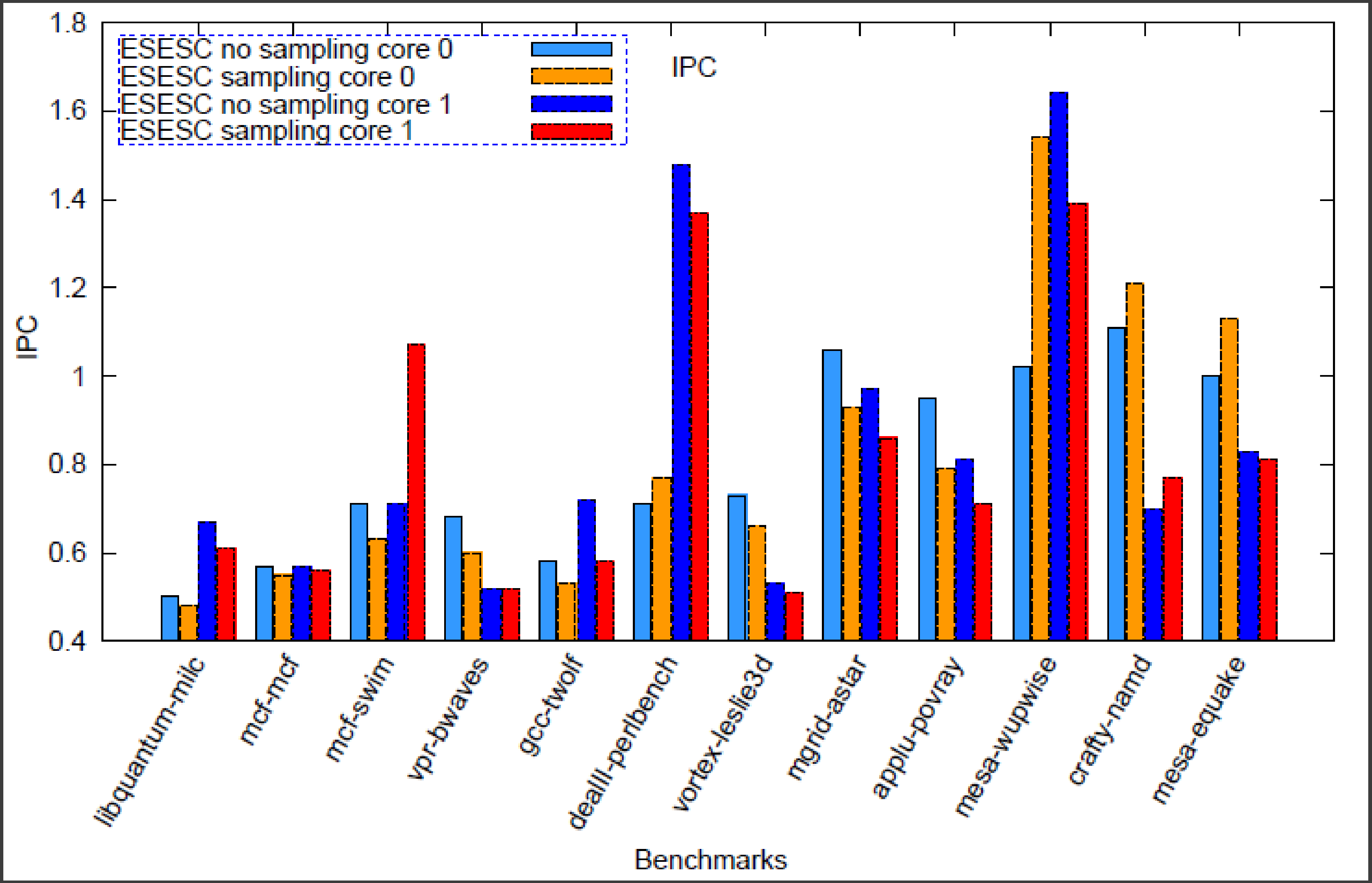
Outline

- Sampling Overview
- ESESC sampling parameters
- Demo single threaded sampling
- Demo multithreaded sampling
- Collecting statistics in ESESC
- Demo adding a new statistics counter

Sampling Overview



Sampling Overview

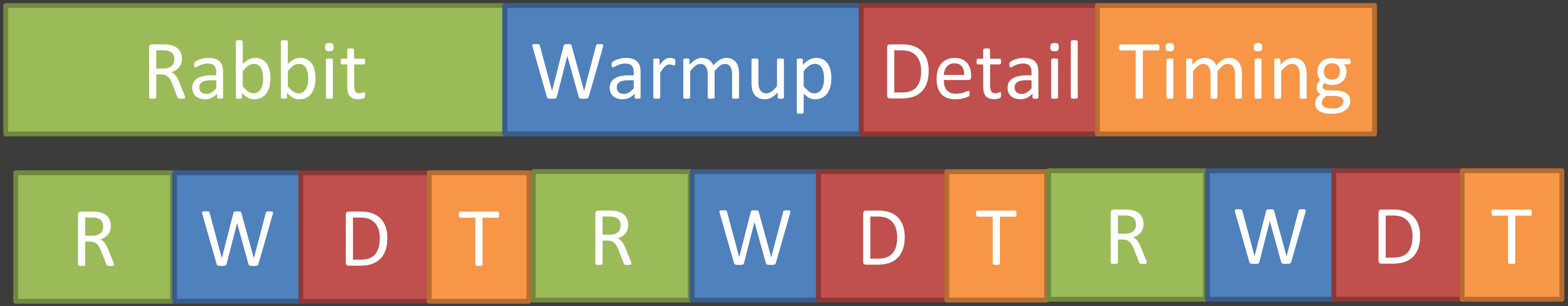


Outline

- Sampling overview
- ESESC sampling parameters
- Demo single threaded sampling
- Demo multithreaded sampling
- Collecting statistics in ESESC
- Demo adding a new statistics counter

ESESC Sampling Modes

- Rabbit
  - Emulation only
- Warmup
  - Update cache
- Detail
  - Full simulation but discard statistics
- Timing
  - Full simulation and keep statistics

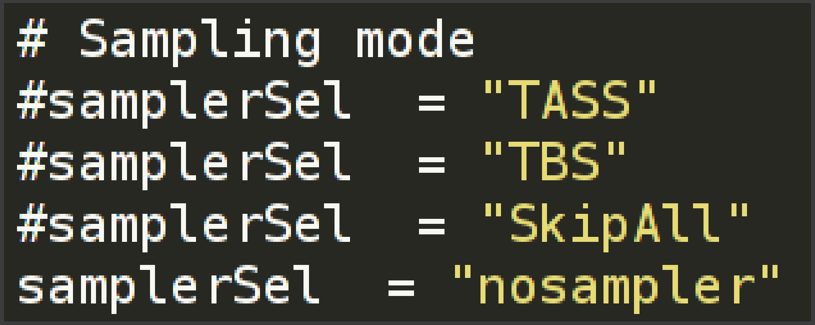


Rabbit Warmup Detail Timing

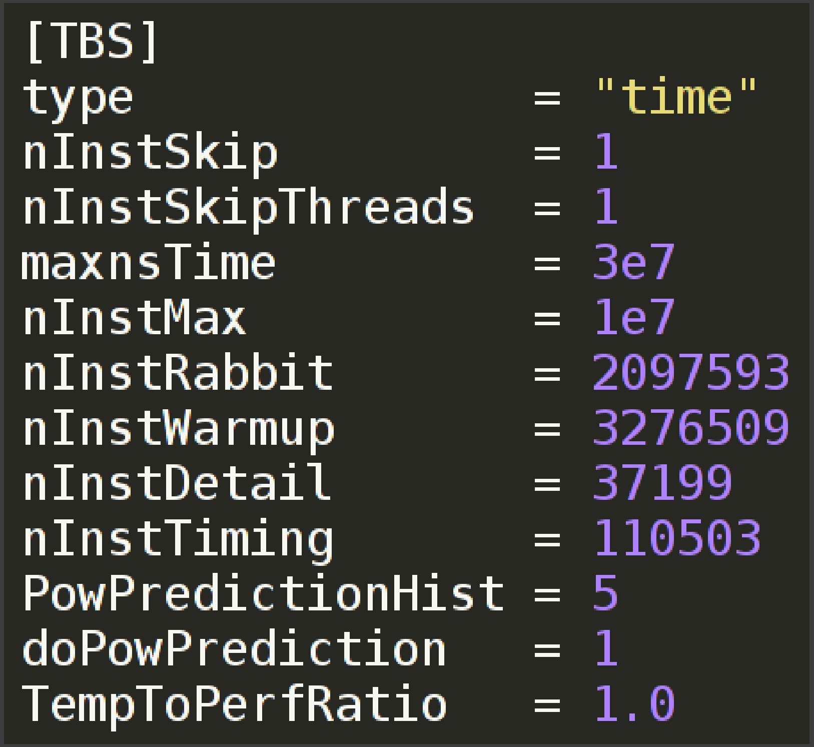
R W D T R W D T R W D T

ESESC Sampling Parameters

- type
  - inst, time
- nInstSkip
  - Number of instructions main thread skips
- nInstSkipThreads
  - Number of instructions spawned threads skip
- maxnsTime
  - Maximum simulation time in ns
- nInstMax
  - Maximum number of instructions to simulate
- nInstRabbit
  - Emulation only
- nInstWarmup
  - Emulation plus cache
- nInstDetail
  - Detailed warm-up of full pipeline
- nInstTiming
  - Collect statistics



```
# Sampling mode
#samplerSel  = "TASS"
#samplerSel  = "TBS"
#samplerSel  = "SkipAll"
samplerSel  = "nosampler"
```



```
[TBS]
type                 = "time"
nInstSkip            = 1
nInstSkipThreads     = 1
maxnsTime            = 3e7
nInstMax             = 1e7
nInstRabbit          = 2097593
nInstWarmup          = 3276509
nInstDetail          = 37199
nInstTiming          = 110503
PowPredictionHist    = 5
doPowPrediction      = 1
TempToPerfRatio      = 1.0
```

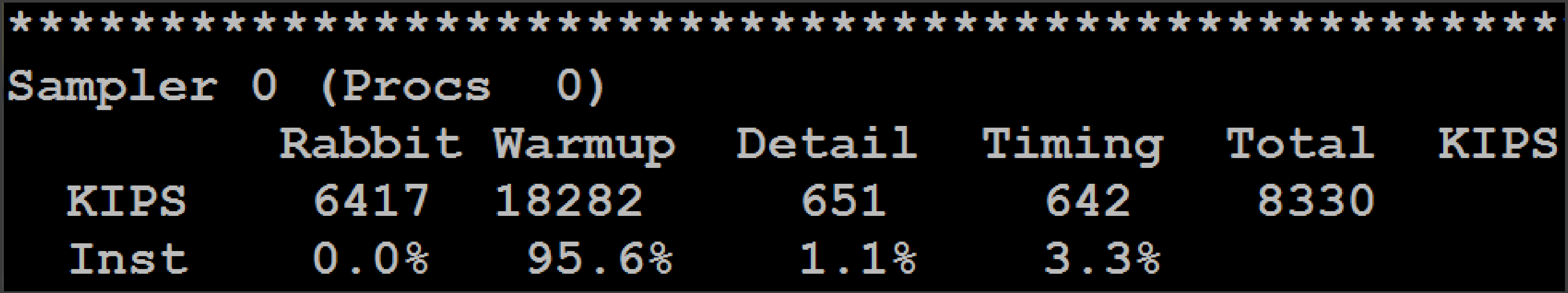
Outline

- Sampling overview
- ESESC sampling parameters
- Demo single threaded sampling
- Demo multithreaded sampling
- Collecting statistics in ESESC
- Demo adding a new statistics counter

Demo 1: Single-threaded with TBS

- Run crafty with TBS
- Single core configured

Sampling Report



```
***************************************************
Sampler 0 (Procs  0)
         Rabbit Warmup  Detail  Timing  Total  KIPS
  KIPS    6417  18282     651     642    8330
  Inst    0.0%   95.6%    1.1%    3.3%
```

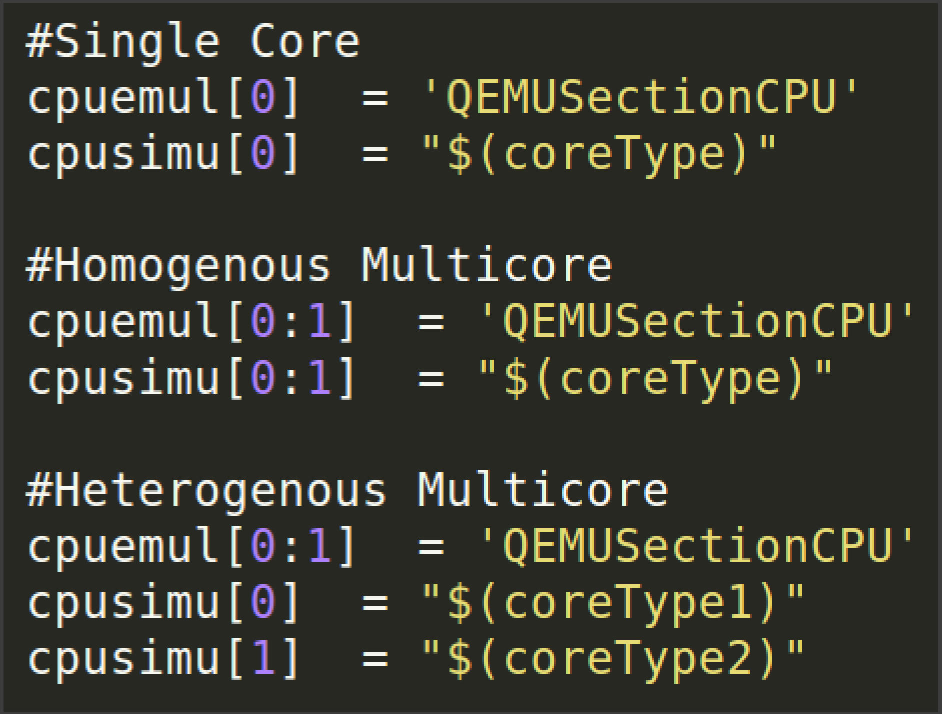
- Instruction percentage and speed in each mode

Outline

- Sampling overview
- ESESC sampling parameters
- Demo single threaded sampling
- Demo multithreaded sampling
- Collecting statistics in ESESC
- Demo adding a new statistics counter

ESESC Multicore Configuration

- Homogenous and heterogeneous multicore configuration
- Homogenous
  - cpusimu[0:NUM_CORES-1]
- Heterogeneous
  - cpusimu[0] = ‘coreType1’
  - cpusimu[1] = ‘coreType2’



```
#Single Core
cpuemul[0]  = 'QEMUSectionCPU'
cpusimu[0]  = "$(coreType)"

#Homogenous Multicore
cpuemul[0:1]  = 'QEMUSectionCPU'
cpusimu[0:1]  = "$(coreType)"

#Heterogenous Multicore
cpuemul[0:1]  = 'QEMUSectionCPU'
cpusimu[0]  = "$(coreType1)"
cpusimu[1]  = "$(coreType2)"
```

Demo 2: Multithreaded Sampling

- Run FFT with TBS
- Multicore configuration

Outline

- Sampling overview
- ESESC sampling parameters
- Demo single threaded sampling
- Demo multithreaded sampling
- Collecting statistics in ESESC
- Demo adding a new statistics counter

Statistics in ESESC

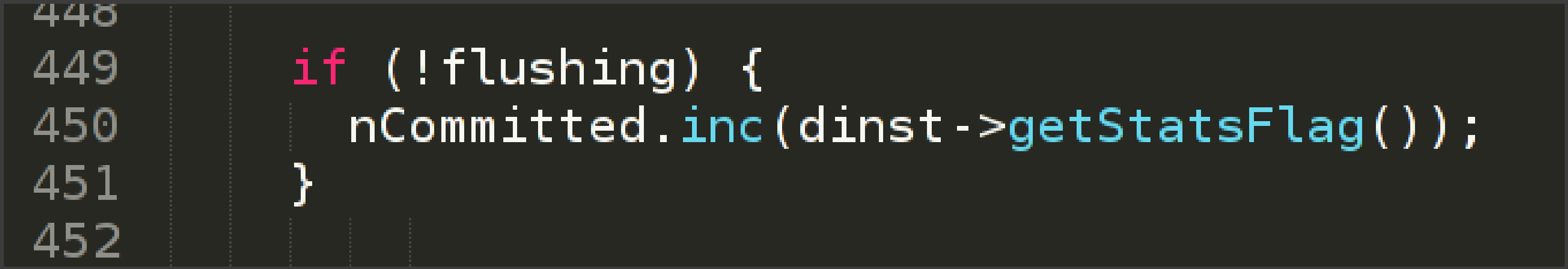
- Unified statistics management with GStats classes
- Raw GStats output is processed by report.pl script
- Only collect statistics during timing simulation

Statistics with Sampling

- Use GStats class for simulation statistics
- Each GStat must have a unique name
  - All stats stored in a hash map
  - Need to use state of instruction when updating counter
- `GStatsCntr`
  - Counter that supports
    - `add` – add specified amount to counter
    - `inc` – increment by 1
    - `dec` – decrement by 1
- `GStatsAvg`
  - Average value
- `GStatsMax`
  - Number of samples and max value

Example GStat: nCommitted

- GStatsCntr nCommitted counts number of committed instructions
  - Defined in GProcessor.h as part of GProcessor class
- Used in OoOProcessor which inherits from GProcessor
- Reads state of dinst->getStatsFlag()



```
    if (!flushing) {
      nCommitted.inc(dinst->getStatsFlag());
    }
```

Outline

- Sampling overview
- ESESC sampling parameters
- Demo single threaded sampling
- Demo multithreaded sampling
- Collecting statistics in ESESC
- Demo adding a new statistics counter

Demo 3: Add a Counter to ESESC

- Add counter to ESESC
- Run and explain GStats output

Summary

- ESESC uses sampling for speed-up
- Sampling has minimum effect in accuracy
- Time-based sampling supports single-threaded and multi-threaded simulation
- Sampling mode and parameters can be easily changed
- ESESC uses Gstats for gathering statistics
- You can easily add other statistics counters

Sampling References

- SMARTS
  - *SMARTS: Accelerating Microarchitecture Simulation via Rigorous Statistical Sampling,* Wunderlich, et al., ISCA 2003.
- Thermal
  - *Thermal-Aware Sampling in Architectural Simulation,* Ardestani, et al., ISPLED 2012.
- Multithreaded (TBS)
  - *ESESC: A Fast Multicore Simulator Using Time-Based Sampling,* Ardestani, et al., HPCA 2013

Questions?