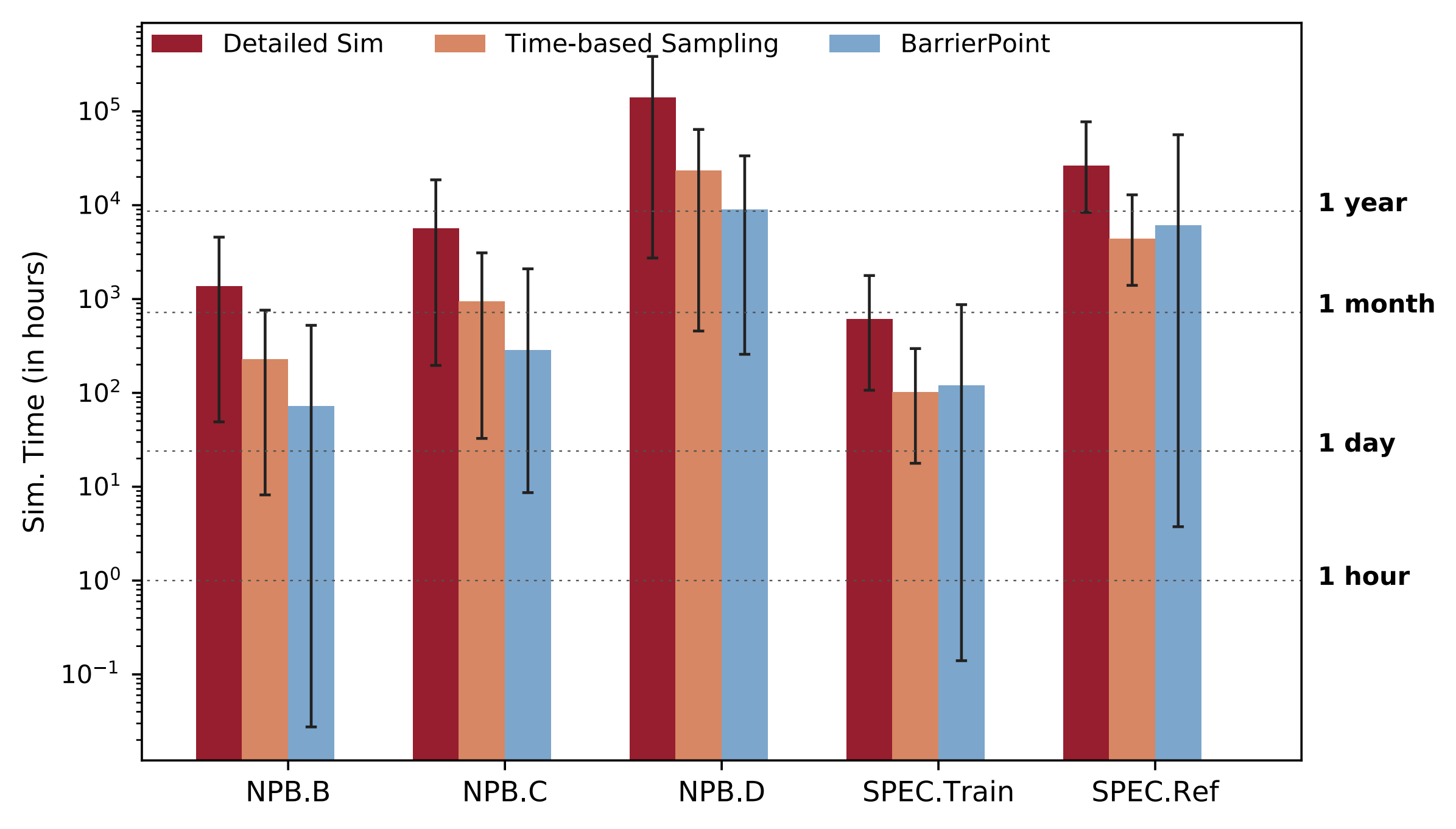
Pac-Sim: Simulation of Multi-threaded Workloads using Intelligent, Live Sampling

Changxi Liu[†], Alen Sabu[†], Akanksha Chaudhari, Qingxuan Kang, Trevor E. Carlson
[†]*Joint first authors*
NUS National University of Singapore

1. Introduction

- Microarchitecture simulation can take months to years
  - *Solution*: Sampled simulation



Threads : eight
Schedule : static
Wait-policy: passive

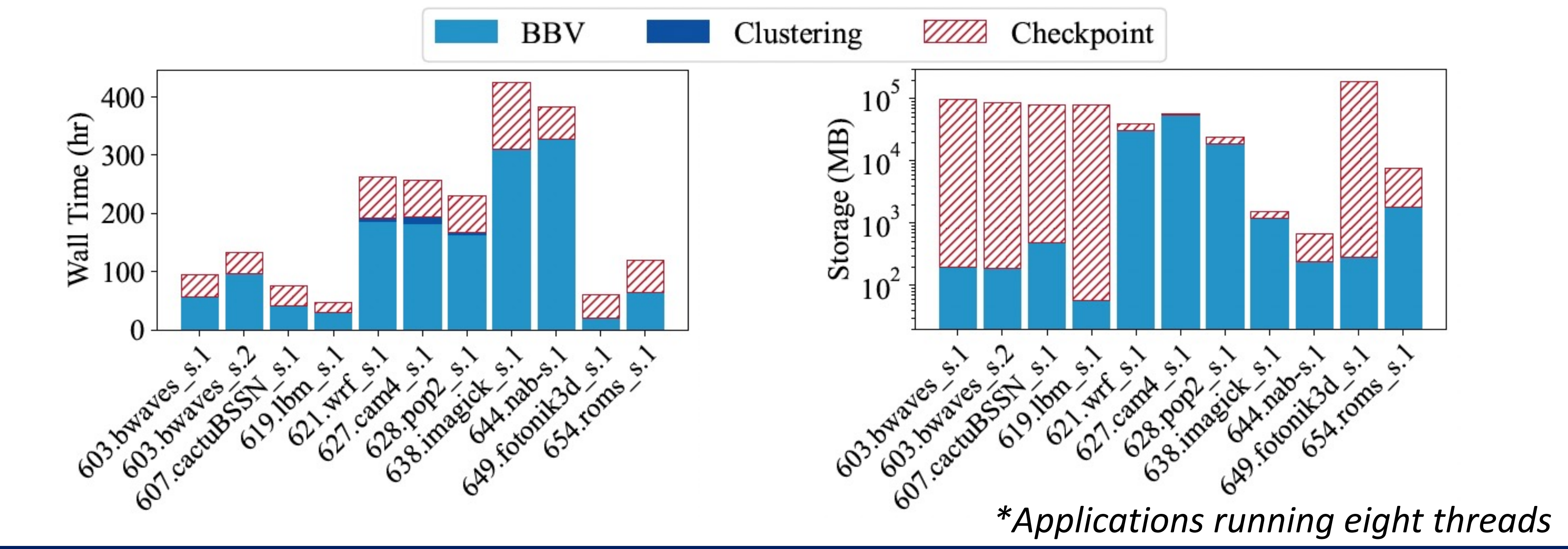
Drawbacks of prior multi-threaded sampling methodologies

Time-based Sampling	BarrierPoint, TaskPoint	LoopPoint
(i) Too slow; defeats the purpose	(ii) Application-specific techniques	(iii) Applies only to static workloads

We propose ***Pac-Sim*** to solve all these problems even in the presence of both hardware and software dynamism

2. Issues with Profile-Driven Techniques

Overheads with LoopPoint Methodology*



**Applications running eight threads*

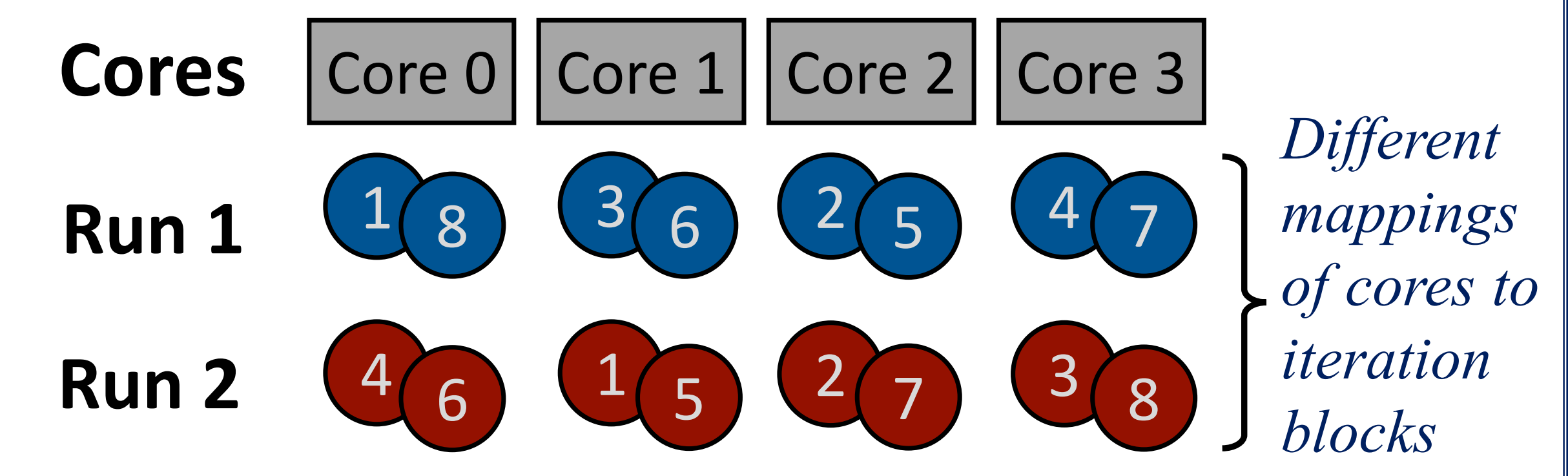
Run-time Optimizations Invalidate Profiling Data

Hardware events

- DVFS
- Turbo Boost
- Cache Reconfiguration

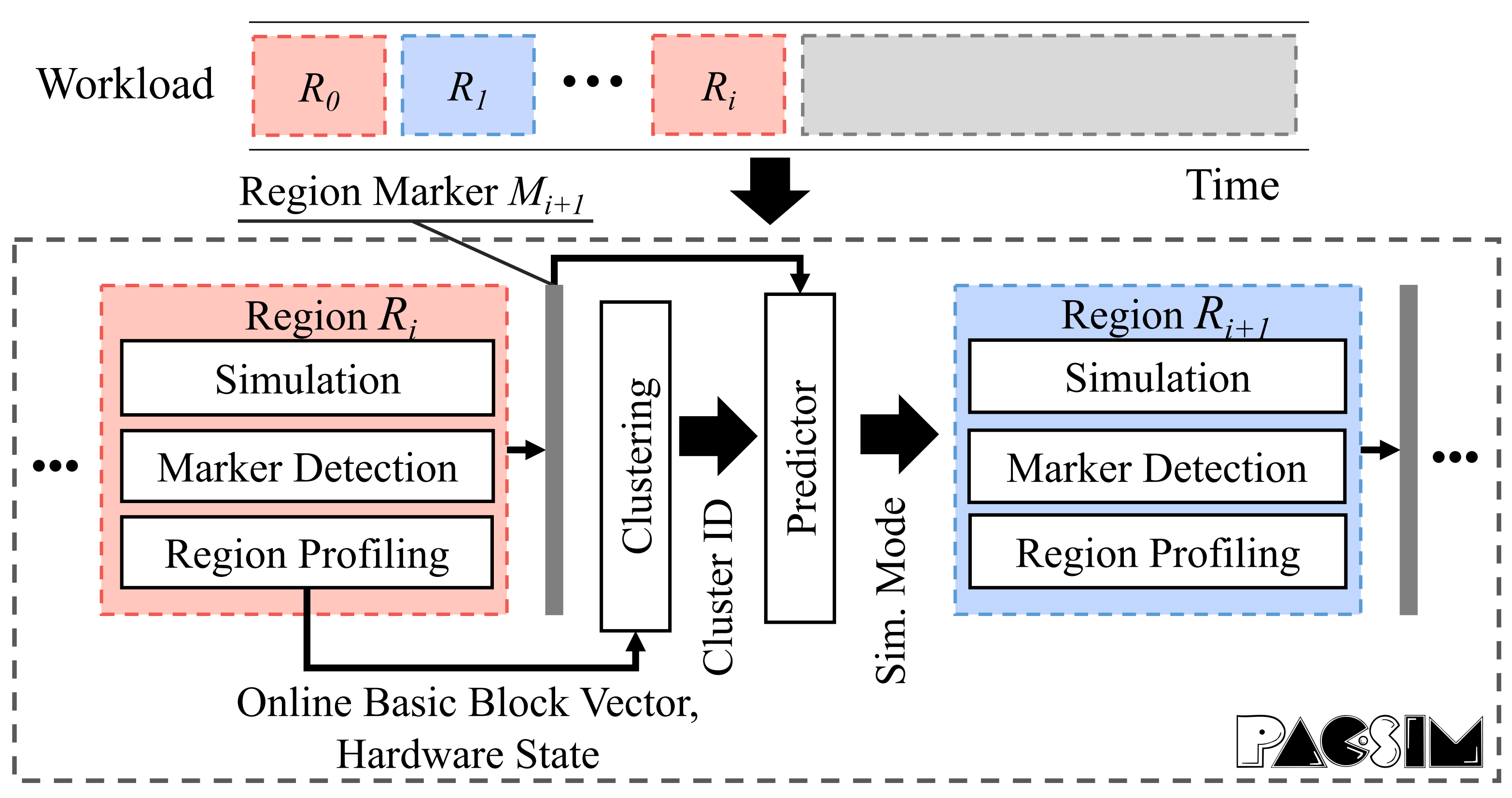
Dynamically scheduled threads

```
#pragma omp parallel for schedule(dynamic)
   for(i=0; i<n; i++) {
      unpredictable_amount_of_work(i);
 }
```



3. *Pac-Sim* Methodology

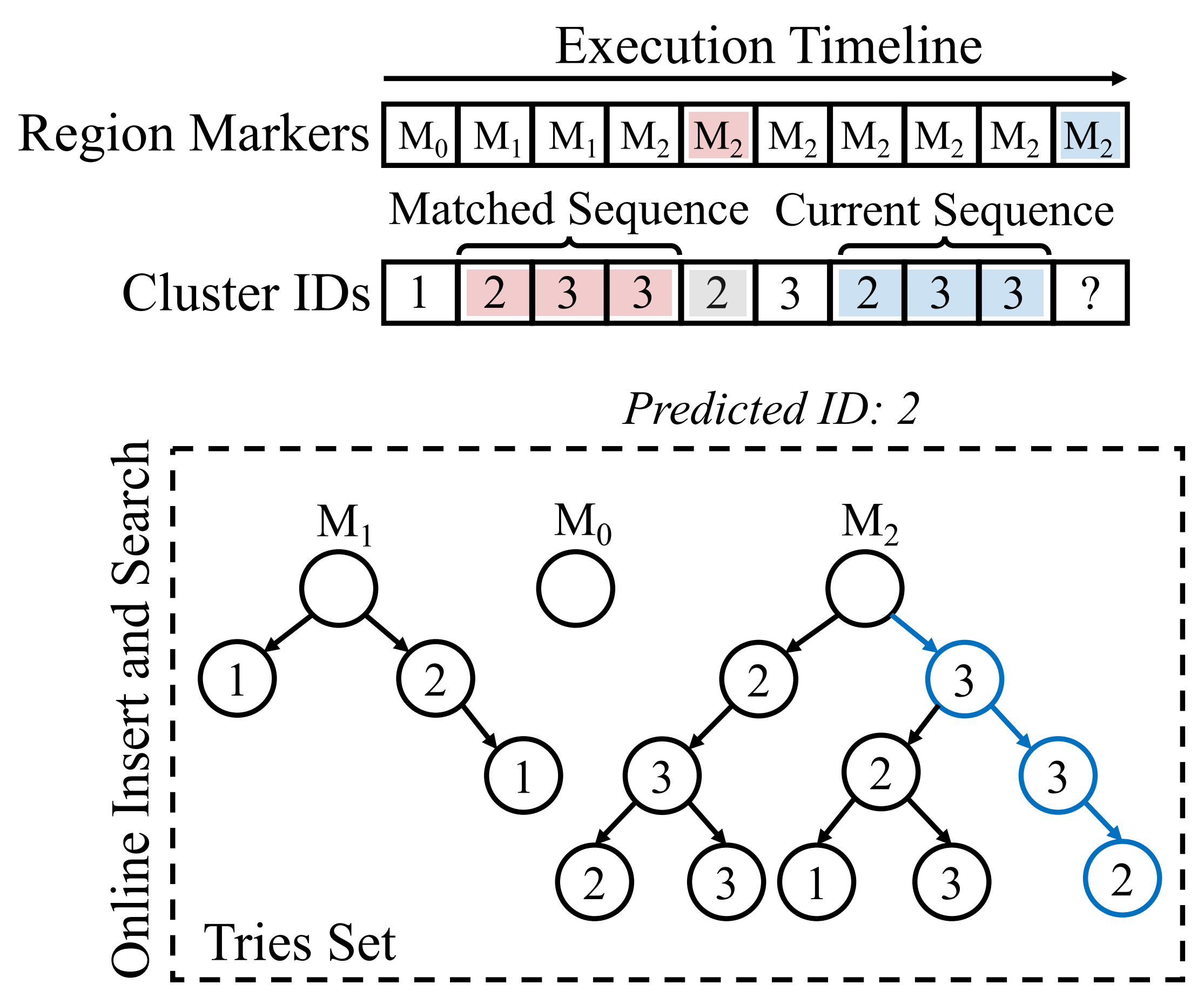
3.1 The Bigger Picture



- Determines simulation regions online which eliminates upfront profiling and checkpointing
- *Marker Detection* identifies the boundaries (as barriers or loops) of the current region
- *Predictor* determines if the next region is to be simulated in detail based on cluster info

3.2 Working of Predictor

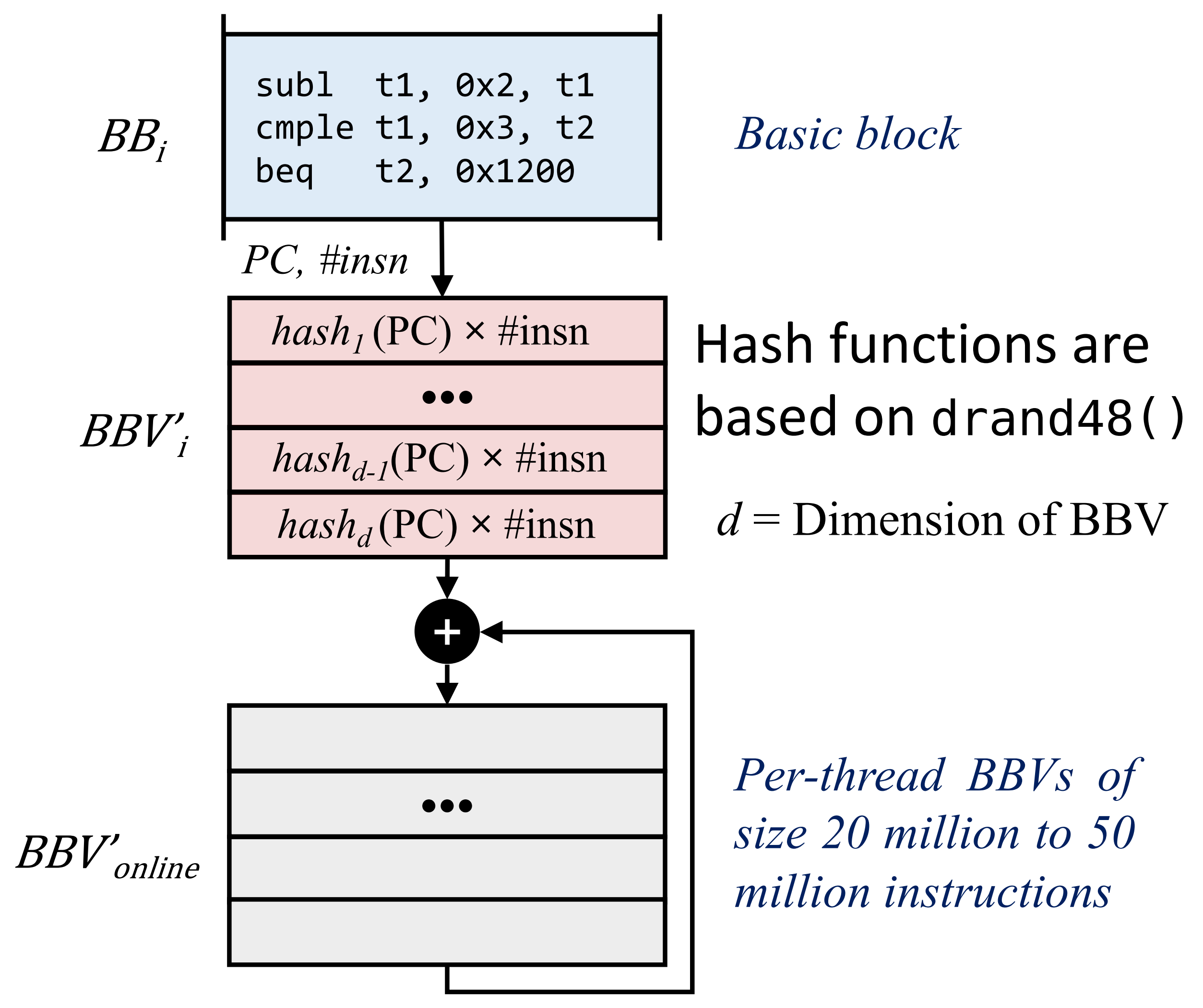
- Cluster history is stored using *tries*



- Next cluster is predicted from longest matching sequence of cluster history
- Pac-Sim uses tries with max-depth of 16

3.3 Online BBVs

- Generated during the simulation of a region



```
subl  t1, 0x2, t1
cmple t1, 0x3, t2
beq   t2, 0x1200
```

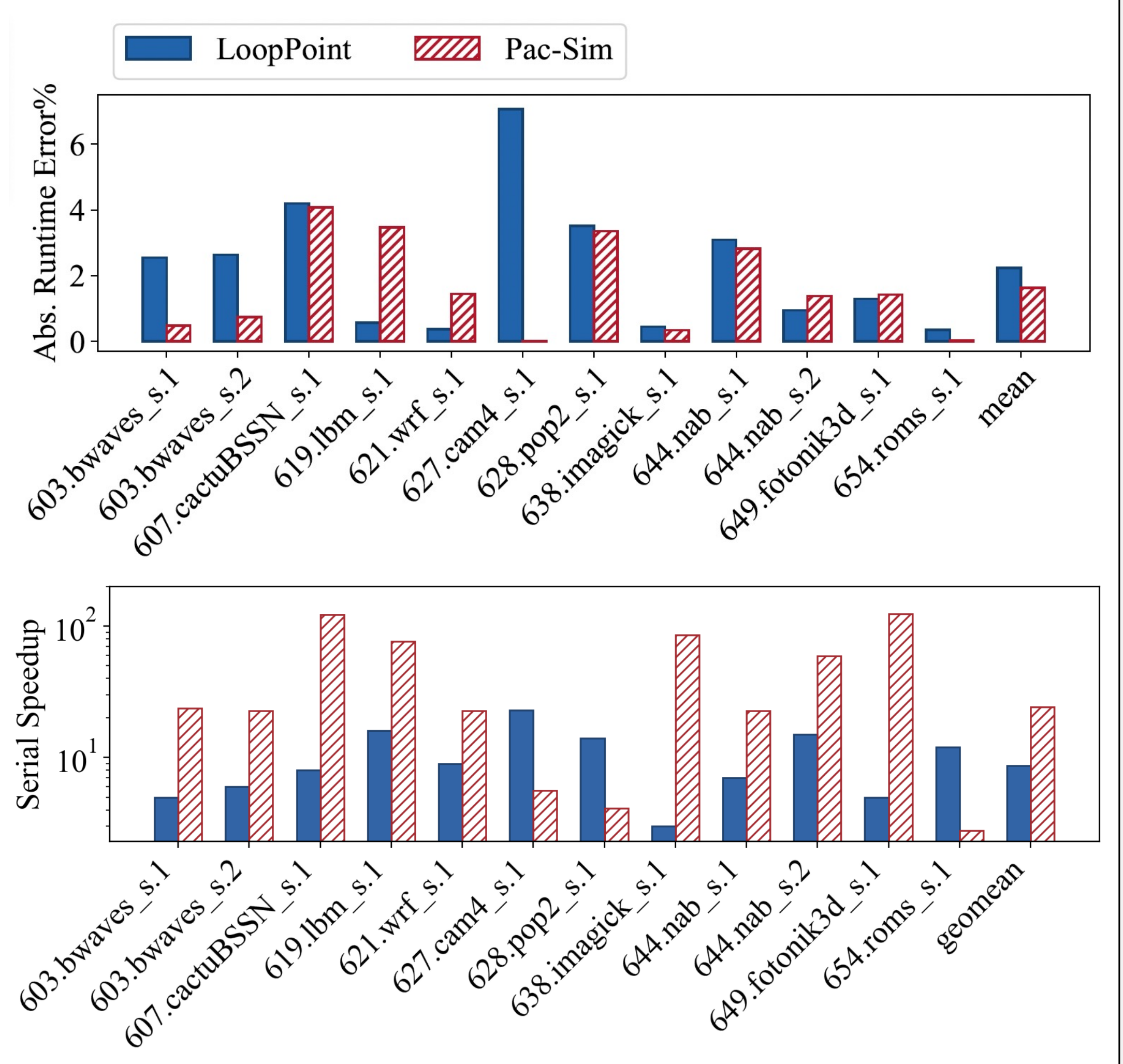
Hash functions are based on `drand48()`

d = Dimension of BBV

$$BBV'_{online} = \sum_i BBV'_i = \sum (BBV_i \,.\, Mproj)$$

4. Experimental Results

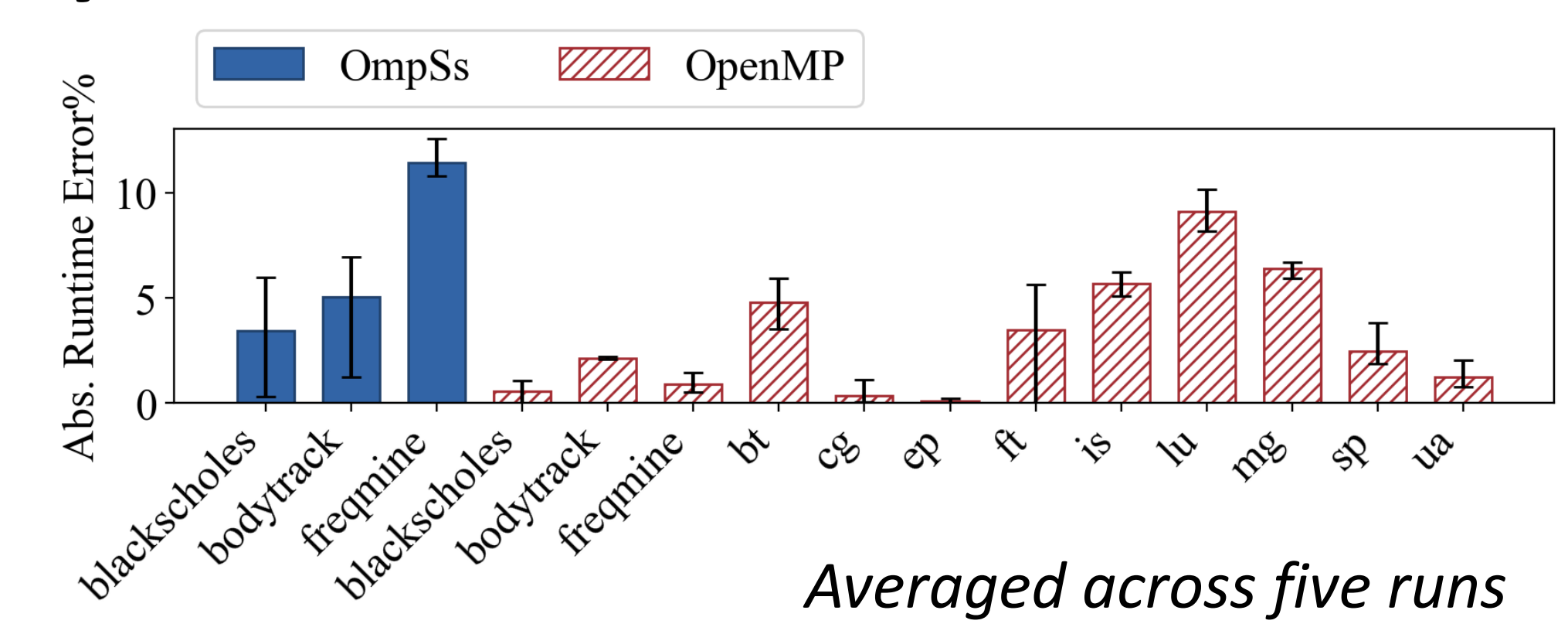
Accuracy and Speedup*



**Applications running eight threads*

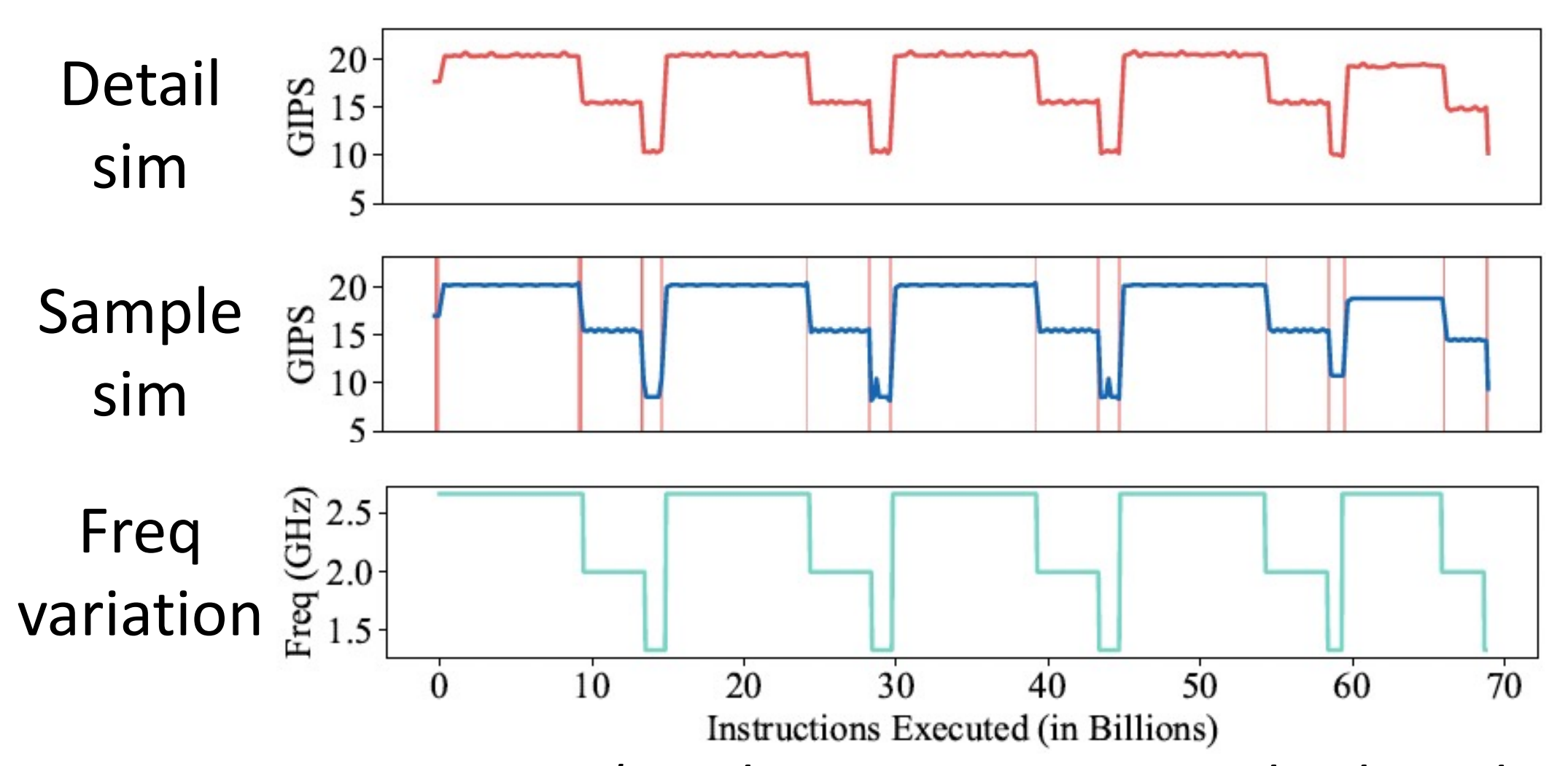
5. Case Study

Dynamic Software*



Averaged across five runs

Dynamic Hardware*



**Applications running eight threads*