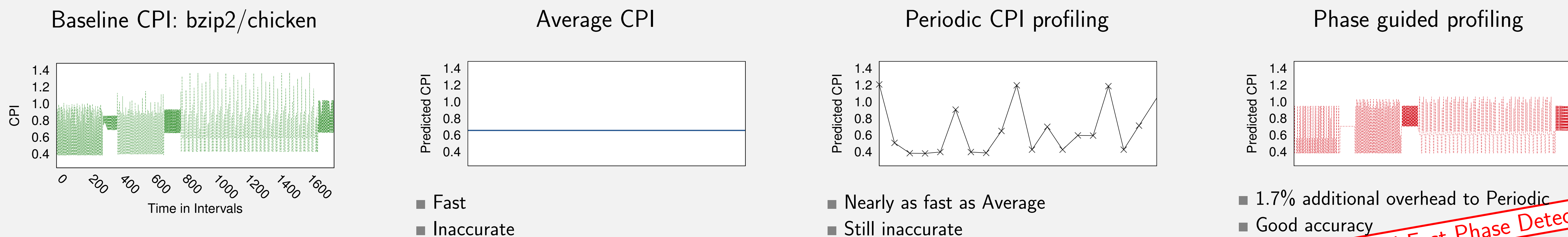


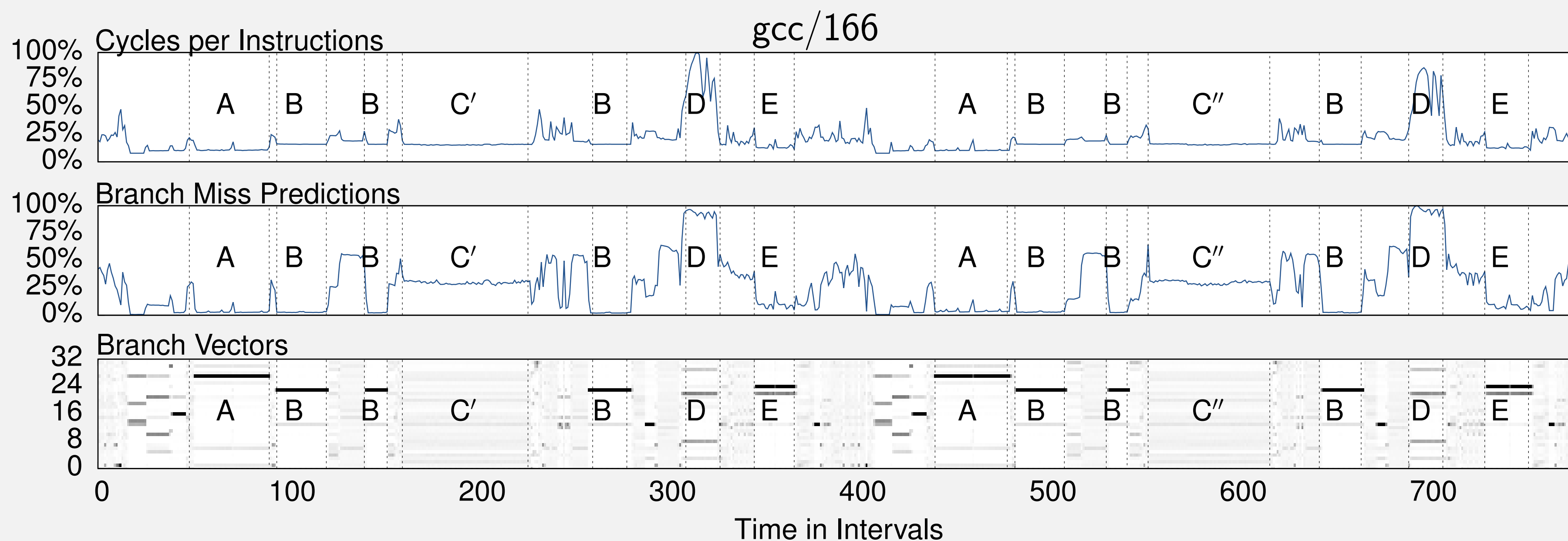
## 1. How can phases help you to profile your application?



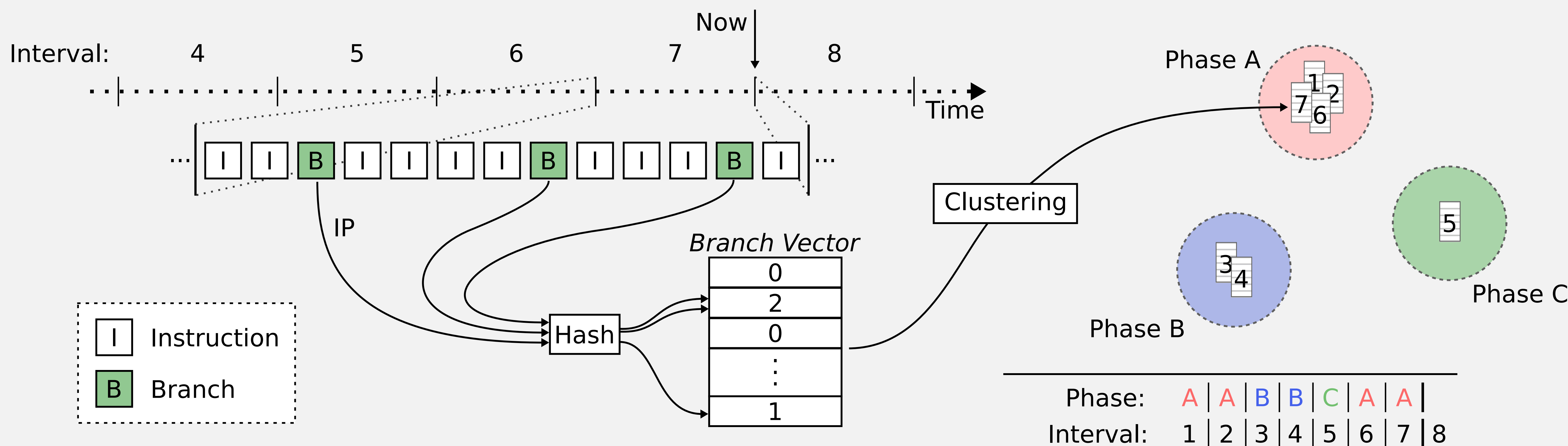
**Need Fast Phase Detection!**

## 2. Why are phases important?

- Most metrics are a function of what code is executed
- Divide the execution into non-overlapping fixed size intervals
- Sample what code is executed
  - BBV - (Dense) Basic Block Vector (Calder et al.)
  - EIPV - Extended Instruction Pointer Vector (Davies et al.)
  - MBBV - Mapped Basic Block Vector (Calder et al.)
  - BRV - Branch Vector
  - CBRV - Conditional Branch Vector
- Intel PEBS to sample basic blocks
- Cluster similar vectors into phases



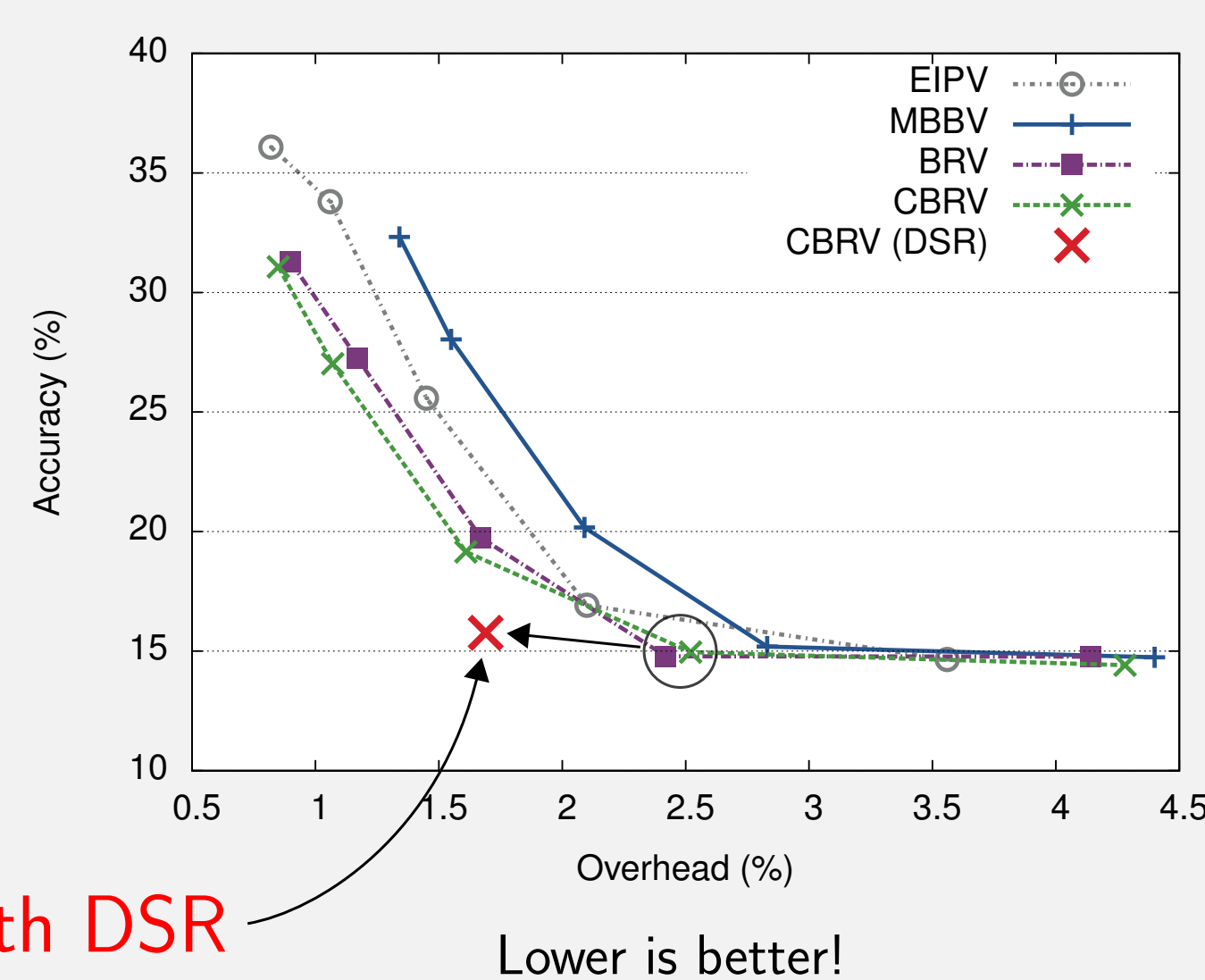
## 3. How to detect phases?



## 4. Accuracy and Overhead

### Code Vectors:

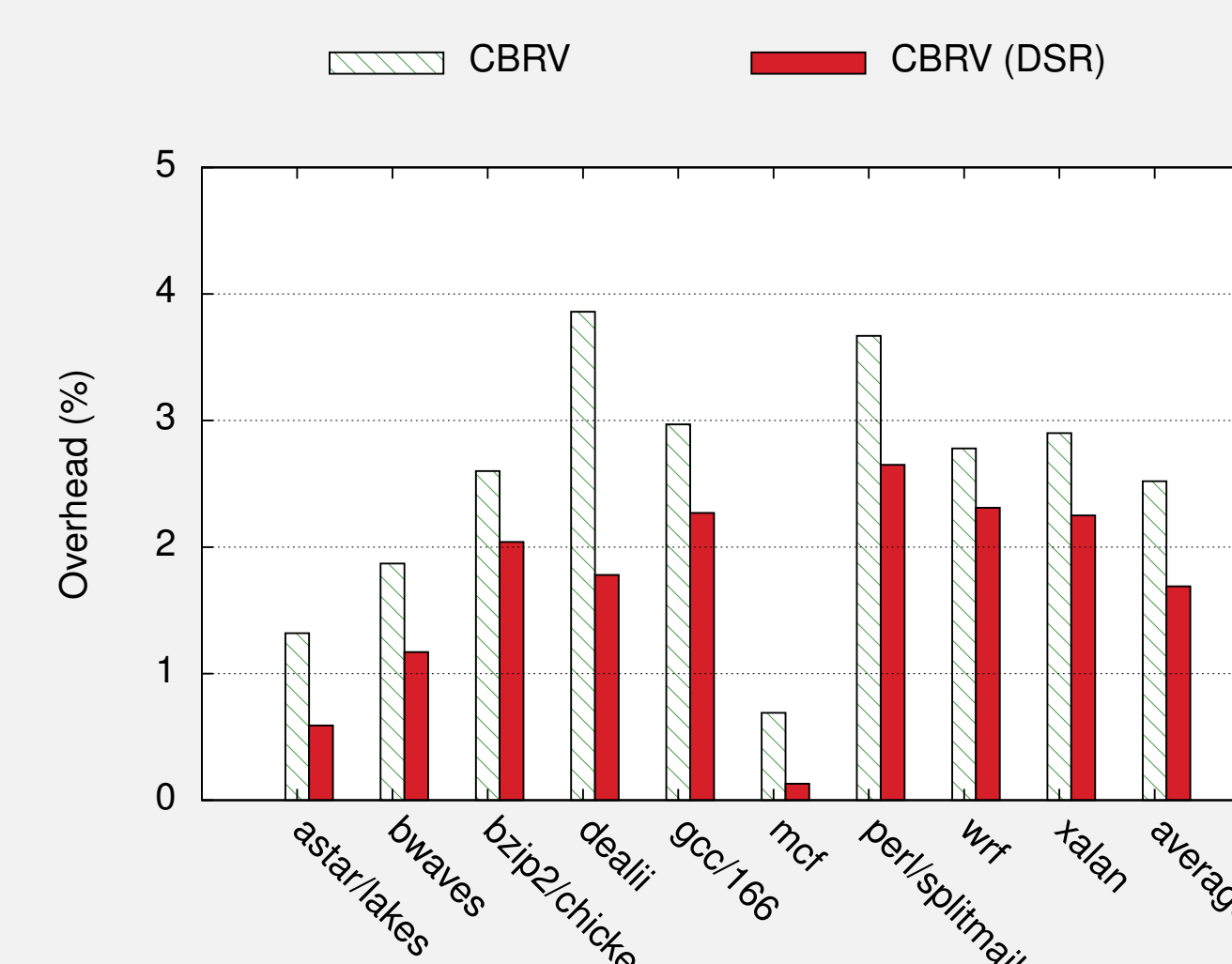
- EIPV has the lowest overhead but the worst accuracy
- MBBV has the highest overhead. It requires the program binary to be parsed to create a IP to basic block mapping.
- BRV/CBRV (this work) shows the best results with respect to accuracy and overhead



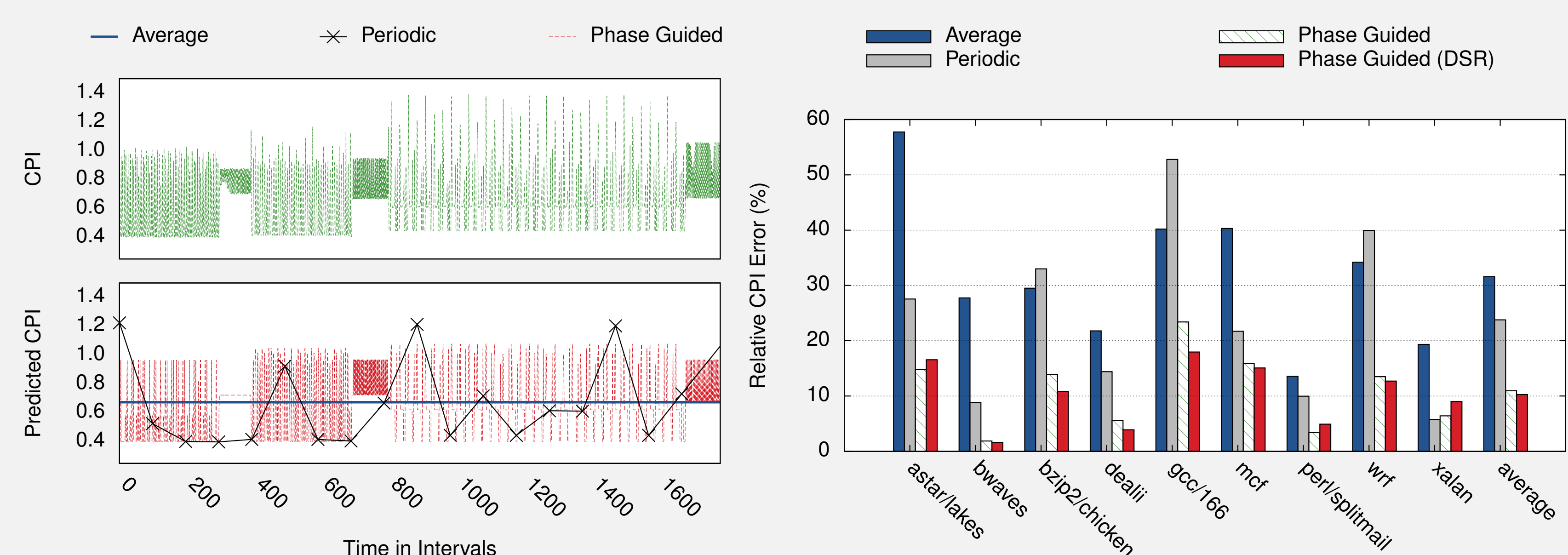
### Dynamic Sample Rate (DSR):

- Low sample rate when the application enters a stable phase
- High sample rate when we predict a phase change
- 33% lower overhead

**1.7% Average Overhead!**



## 5. Use Case: Phase Guided Profiling



- Only profile a small part of each phase
- < 1% of the execution is profiled
- The profiled phases covers 90% of the execution
- Predict the phase in the next interval:
  - Profile the next interval if the phase has not been profiled
  - Otherwise, turn off the profiler
- Each interval is assigned the profile of the phase it belongs to

- 3x better accuracy compared to average profile
- 2x better accuracy compared to periodic profiling
- 1.7% overhead with ScarPhase