Beyond Time-Quantum: A Basic-Block FDA Approach for Accurate System Computing Performance Estimation

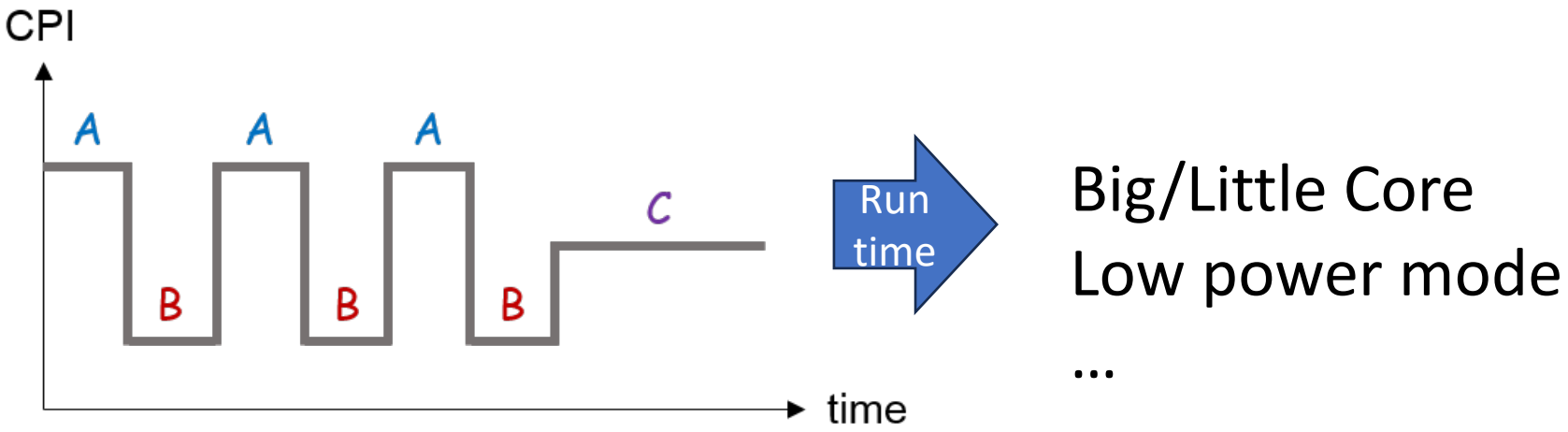
Hsuan-Yi Lin, Ren-Song Tsay

Logos Lab, National Tsing-Hua University

1/25/2024

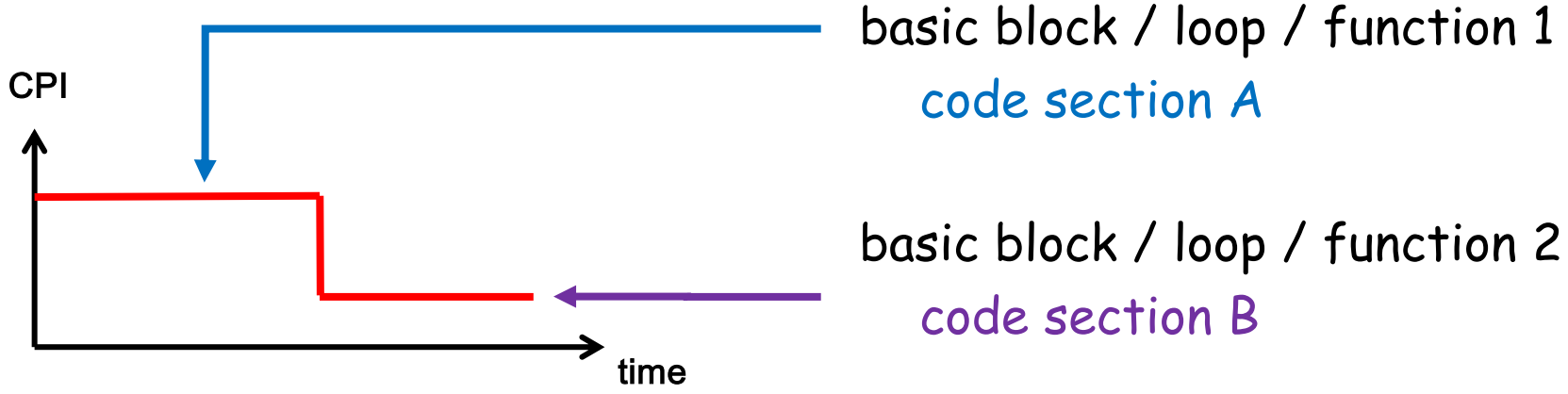
Program Phase – A Sequence of Executions with Similar Behavior

- Most applications have repetitive execution phases which alternatively occur in each execution run.



Highly Correlated to the Program Code Structure

- Application behaviors follow closely with the code functions executed.
- Basic block, loop, and function are common code structures, which usually exhibit stable behavior in repeated executions.



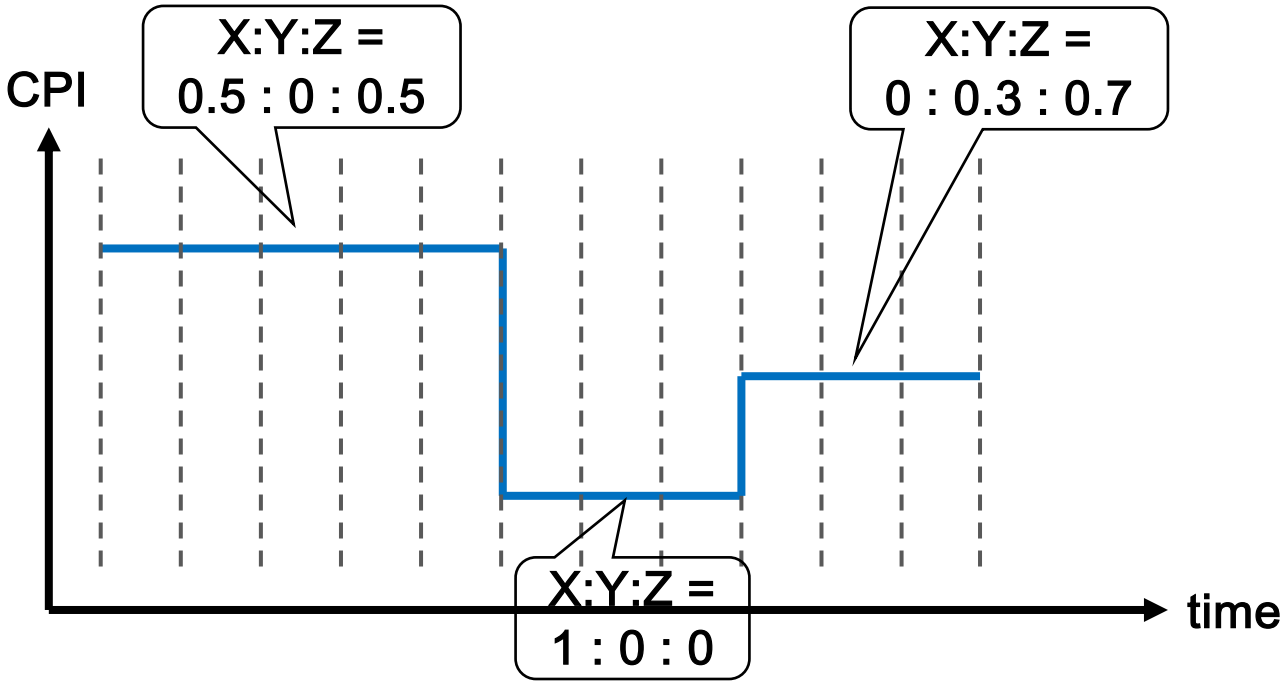
A Few Well-Known Approaches

- Time-quantum-based approach [1][2][3]
  - Divide a program execution into fixed-length intervals, with each interval representing a phase segment that is to be merged into a phase.
- Program-structure-based approach [4]
  - Use the basic structures such as loops or functions to determine program phases.

[1] Phase tracking and prediction, 2003, ISCA
[2] Transition phase classification and prediction, 2005, HPCA
[3] Improving dynamic prediction accuracy through multi-level phase analysis, 2012, LCTES
[4] Selecting Software Phase Markers with Code Structure Analysis, 2006, CGO

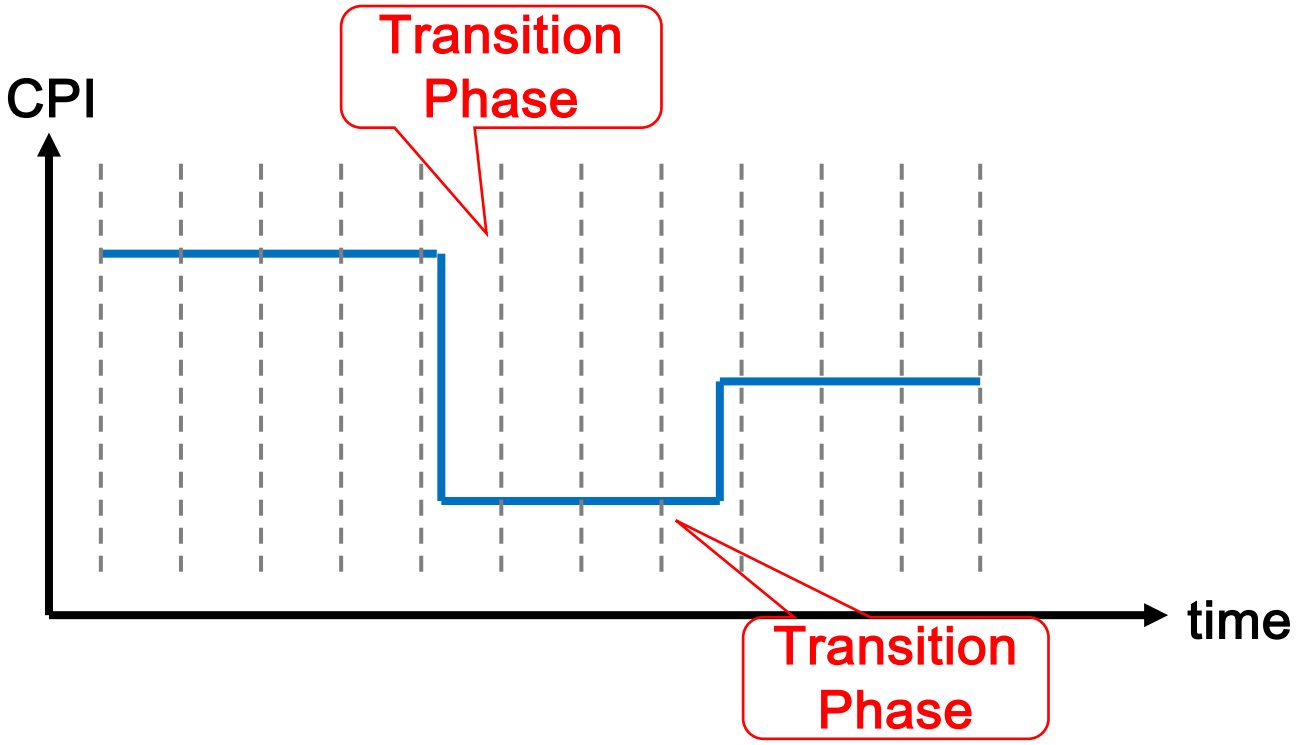
Basic Block Vector Time-Quantum Approach

- Record basic block trace from the execution
- Group time quantum with similar basic block compositions (signatures) into a phase



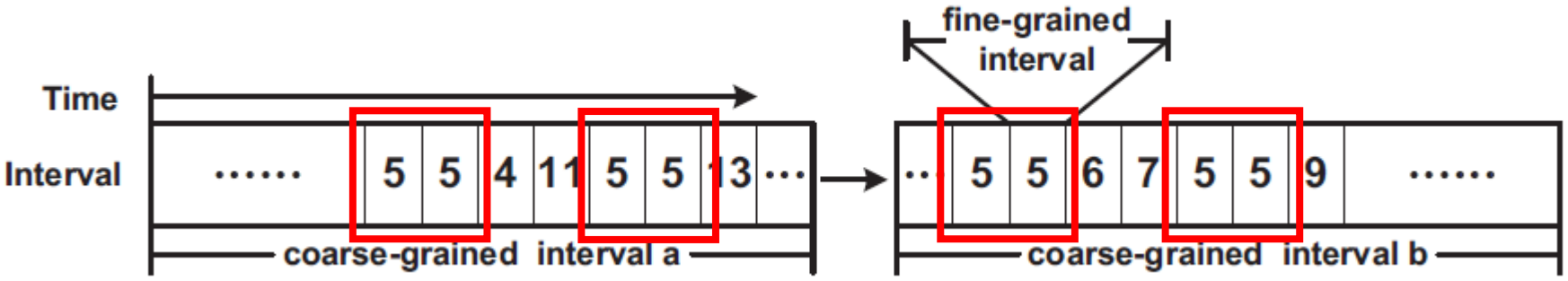
Transition Phase Issue

- A quanta often runs across program phase boundaries



Multi-level Phase Structures

- Phases should be of a hierarchical structure
- Coarse-grain to fine-grain phases.

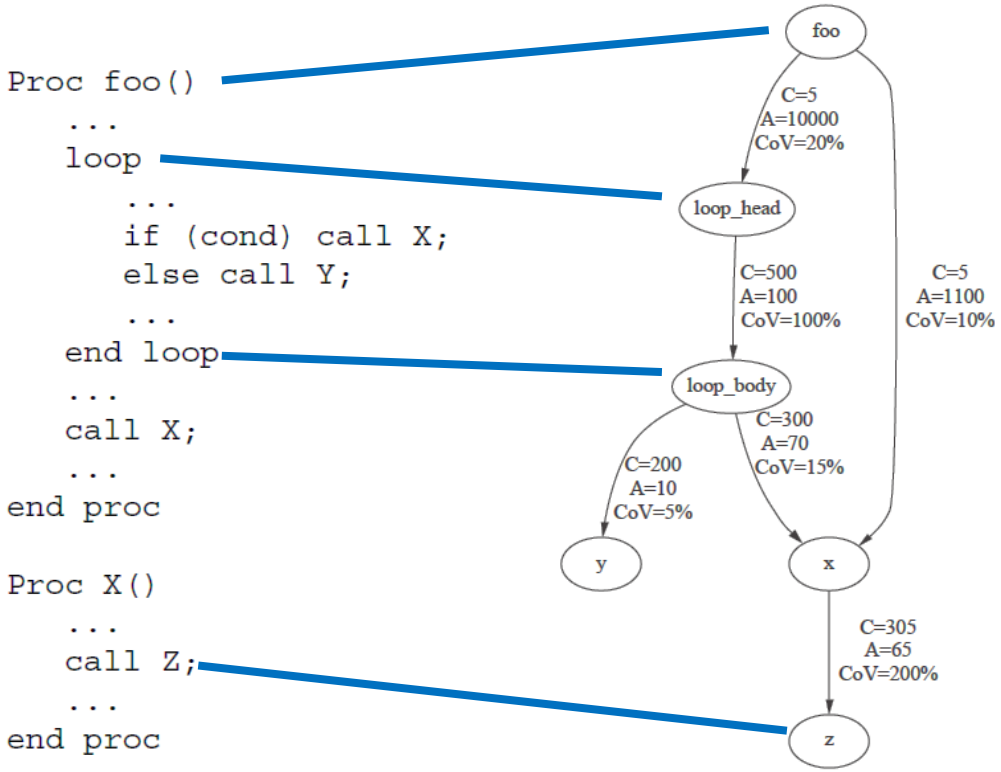


Time-Quantum-Based Approach

- Pros:
  - Convenient for metric evaluation and processing.
- Cons:
  - Cannot precisely capture actual program phases.
  - No insights into the dynamic nature of real program phases.

Program-Structure-Based Approach

- Lau. et. al. used a Hierarchical Call-Loop Graph analysis technique to identify program phases from loops or functions (procedures).

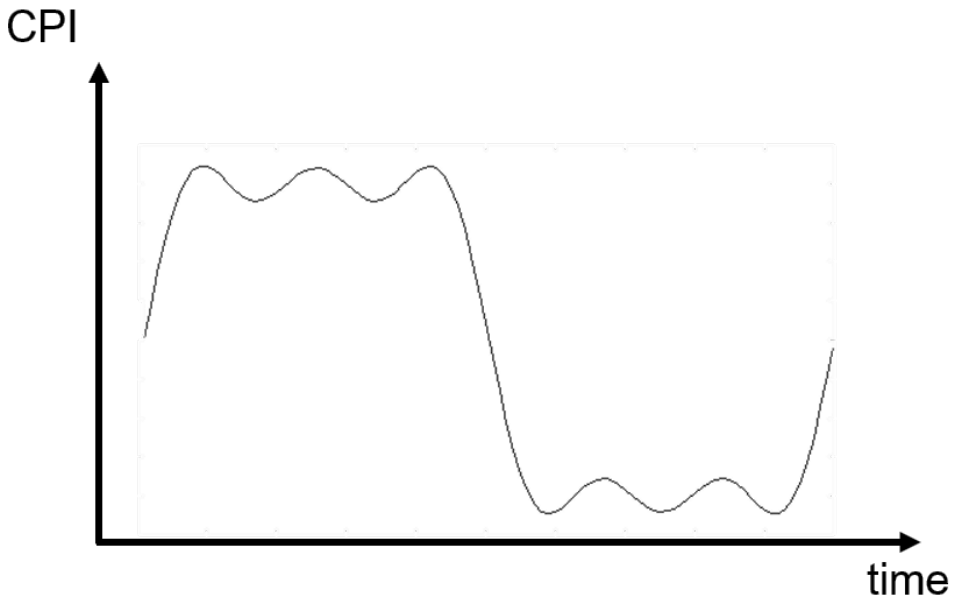


Program-Structure-Based Approach

- Pros:
  - Better understanding of program phases.
  - Stronger associations between each program phase and its characteristics.
- Cons:
  - Difficult to calculate a precise performance value (e.g. CPI) due to the variations of different execution runs.

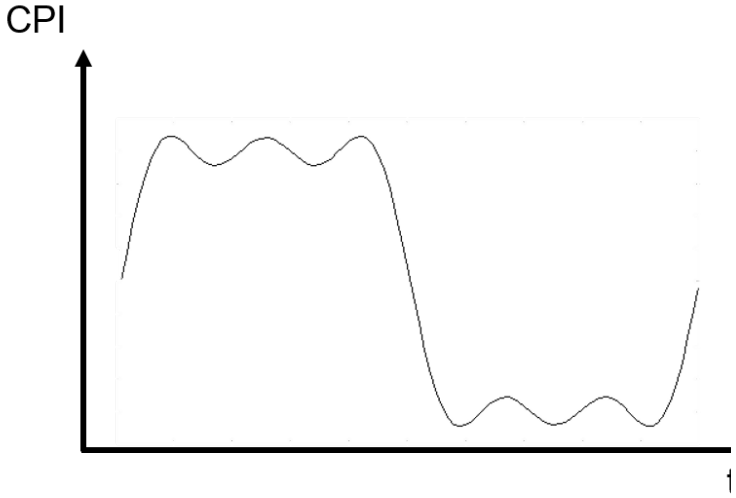
Motivation: Human Intuition in Identifying Program Phases

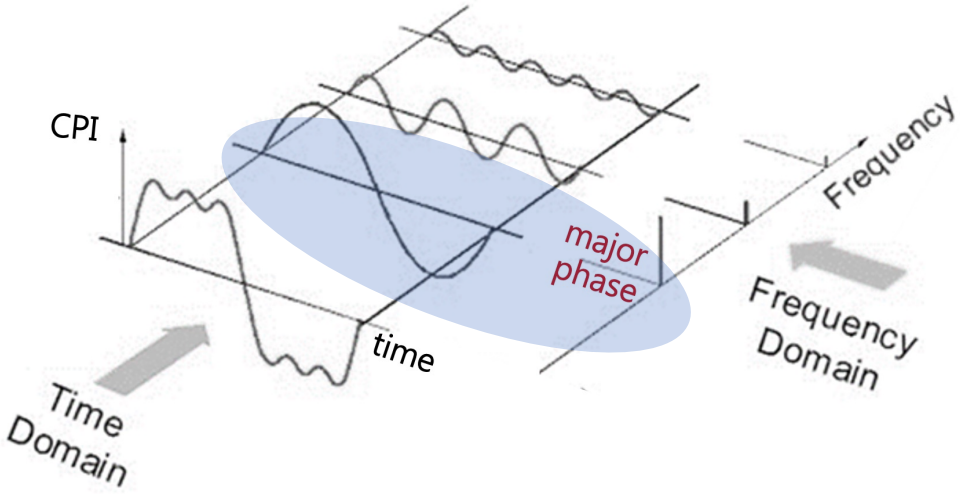
- Simply view the waveform.
- Need a method to systematically and automatically identify these patterns



From Time-domain Waveform to Frequency-domain Spectrum

- The major phase generally dominates the performance waveform.
- Can identify major phase identification through the low-frequency spectrum.

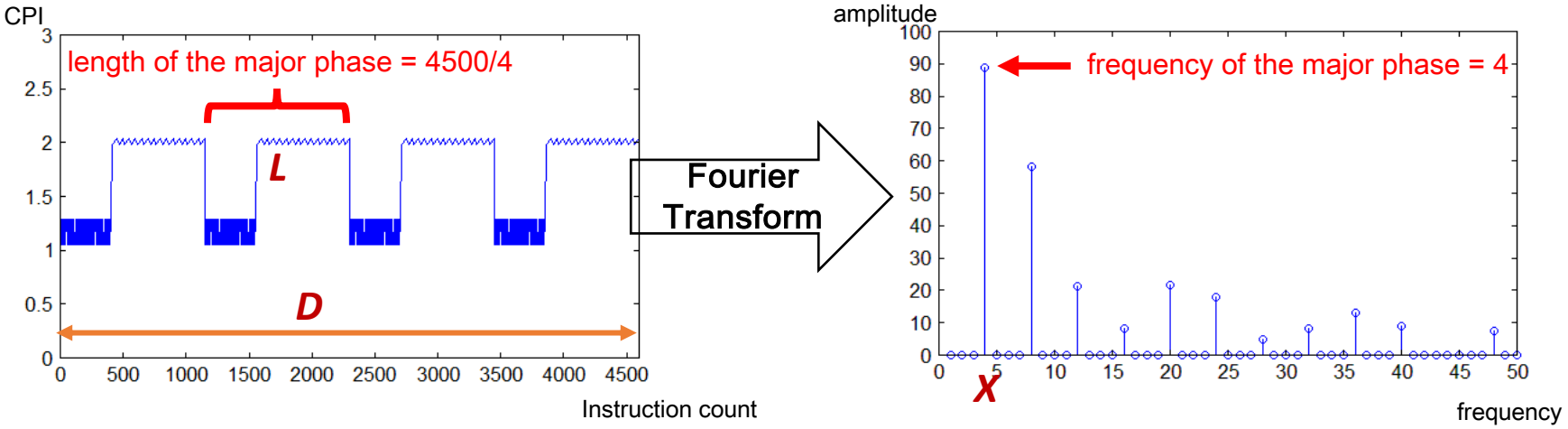




Identify the Main Program Phase

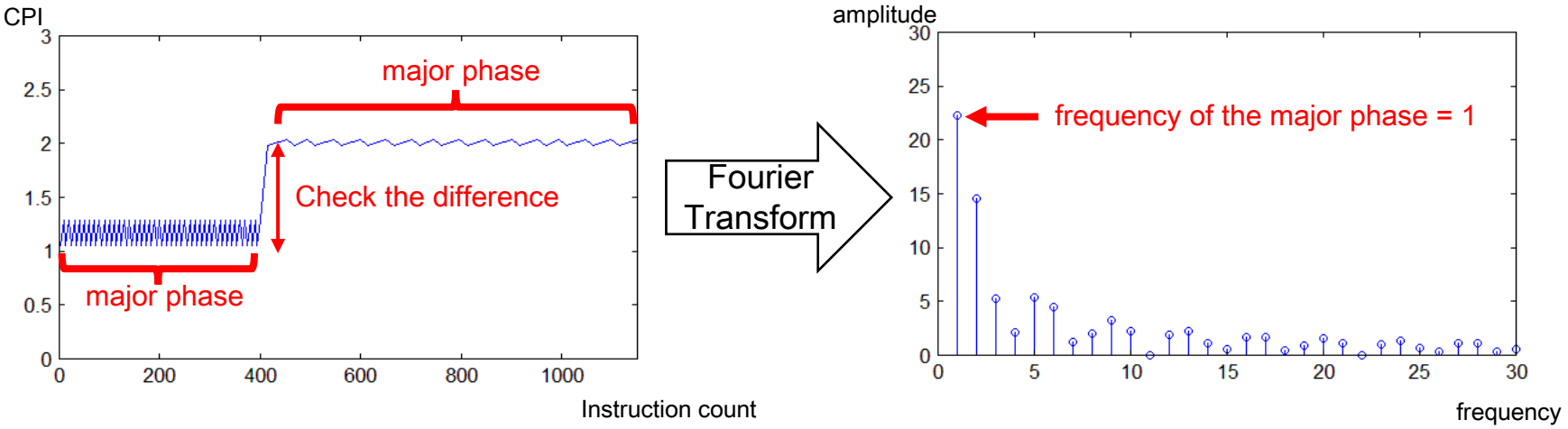
- *D* the total length of the execution length
- *X* the occurrence number of the main spectrum
- *L* the length of the main program phase

$$L = D / X$$



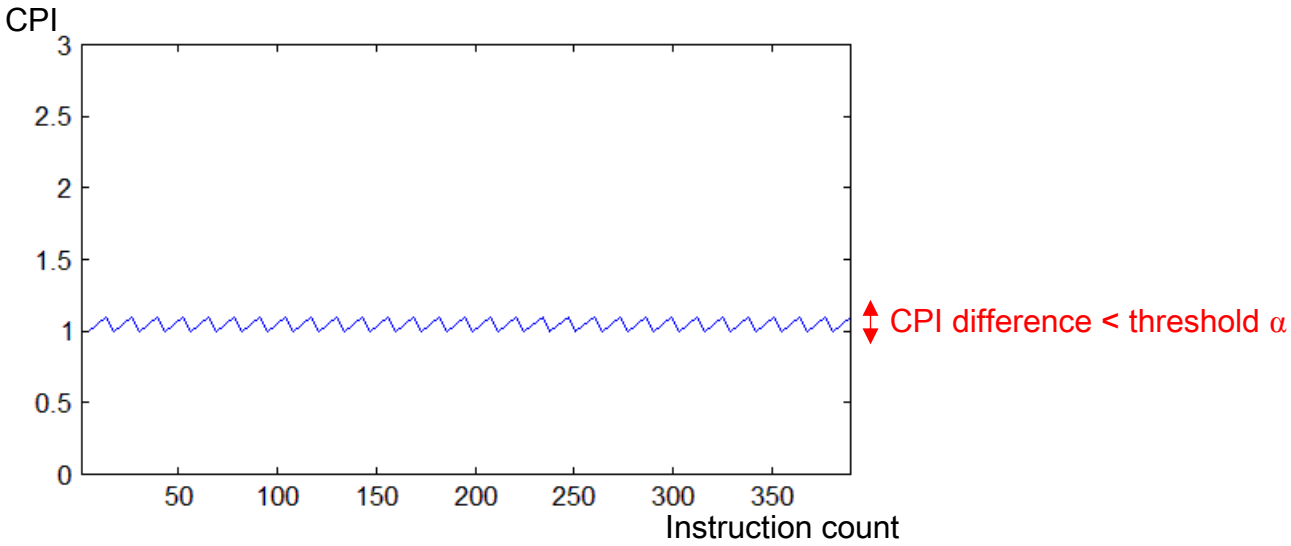
Continue Next Level Analysis Until the Major Phase's Occurrence is One

- No Repeated Pattern
- Check the performance gap to identify the high-low phases.



Stop if Dominant Frequency = Zero

- The time-domain waveform is almost flat
- Only one single phase.

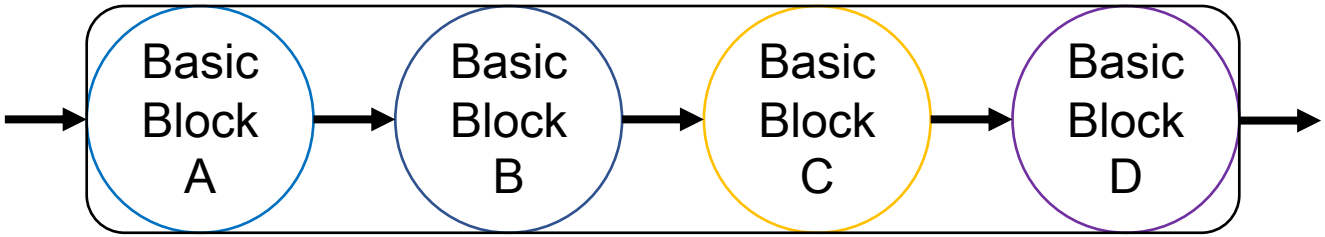


BBFDA Algorithm Summary

1. Profile the target program (by execution or simulation).
2. Apply frequency-domain analysis on the time-domain waveform to generate the frequency spectrum.
3. Calculate the phase length by the main spectrum.
4. Scan the profiling trace to find basic blocks that match the phase length.
5. Check if there are minor phases in this major phase.
6. If yes, go to step 2 and work on the identified major phase; otherwise, the process is completed.

Note: Program Phases Can Be Fully Decomposed into Basic Blocks

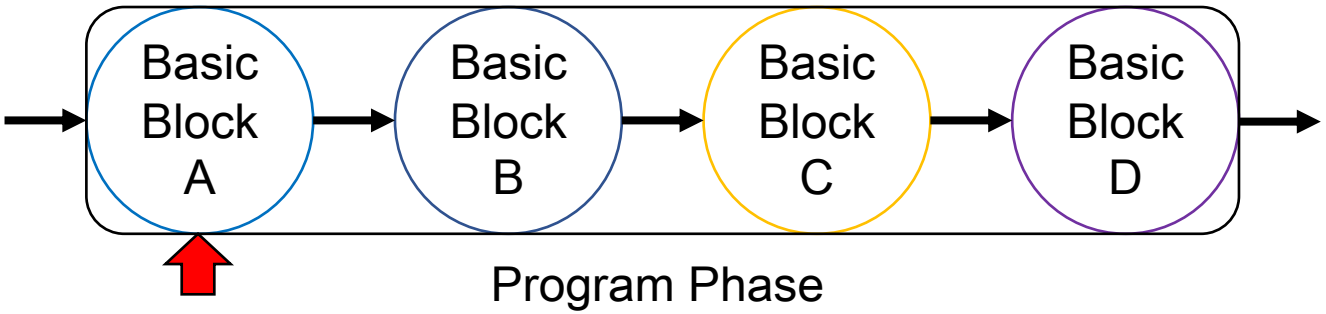
- The transition point between any two program phases is always a branch instruction.
- There are no branch instructions within a basic block.
  - The number and types of instructions are fixed, resulting in consistent performance measures.



Program Phase

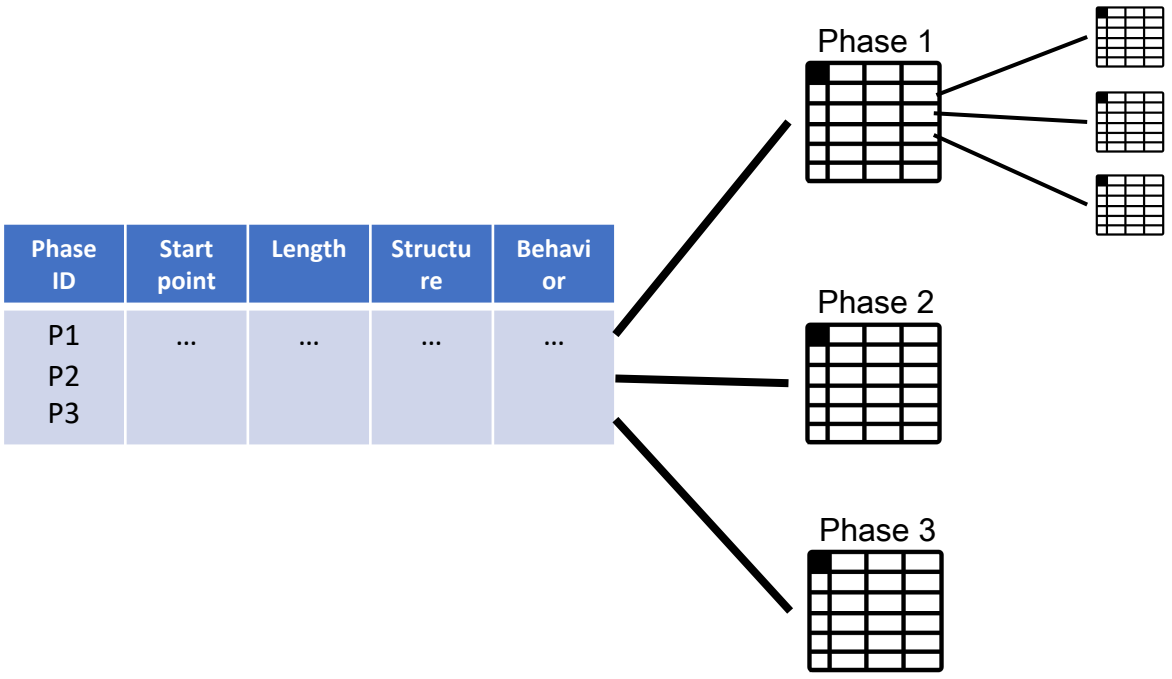
For System Optimization: Identify Phase Starting Basic Block

- Follow branch instructions during program execution to identify basic blocks.
- For each program phase detected, identify the phase starting basic block.



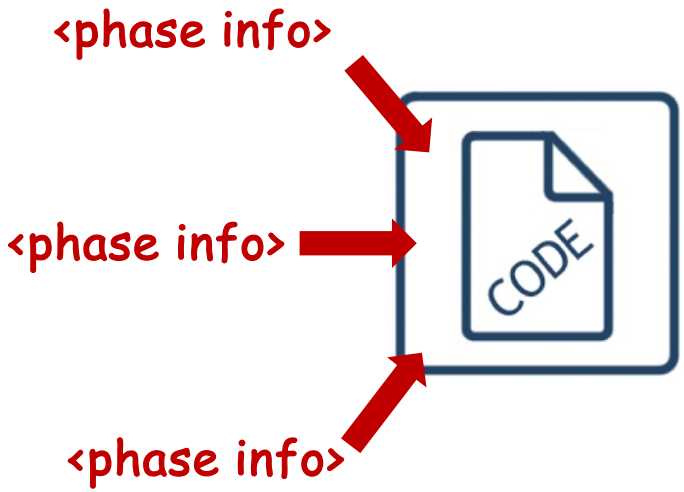
1. Hierarchical Phase Information Table

- Record the phase's starting basic block, length, and performance value.



2. Use a Special Instruction for Hierarchical Phase Information

- Embedded in basic blocks
- Functions like an NOP
- Contains the associated phase information



Validation

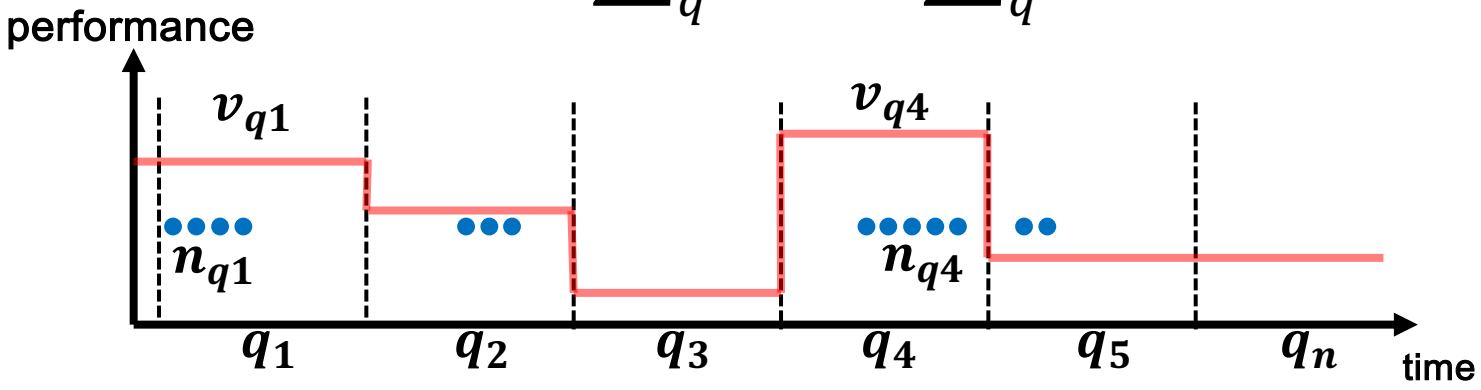
- SPEC CPU2017
- SimpleScalar
- Comparisons

BBFDA	the proposed approach
TQ-100M	the time-quantum-based approach with 100 million instructions quanta size
TQ-1B	the time-quantum-based approach with 1 billion instructions quanta size

Profile the Performance Value p_b of a Basic Block *b* from Time Quantum

- Compute the weighted average from the performance value v_q of the time quanta q where the basic block b occurs n_q times.

$$p_b = (\sum_q n_q v_q)/(\sum_q n_q)$$

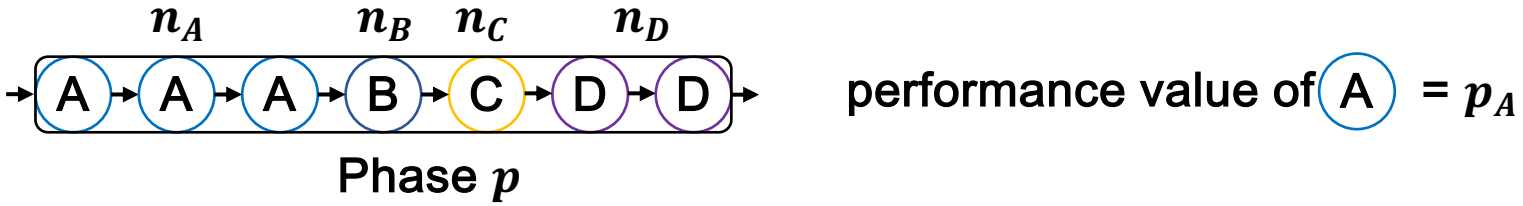


- The estimation matches well with real value.
- Can be further improved using the identified phases.

Estimate the Performance Value p_p of a Program Phase *p* from Basic Blocks

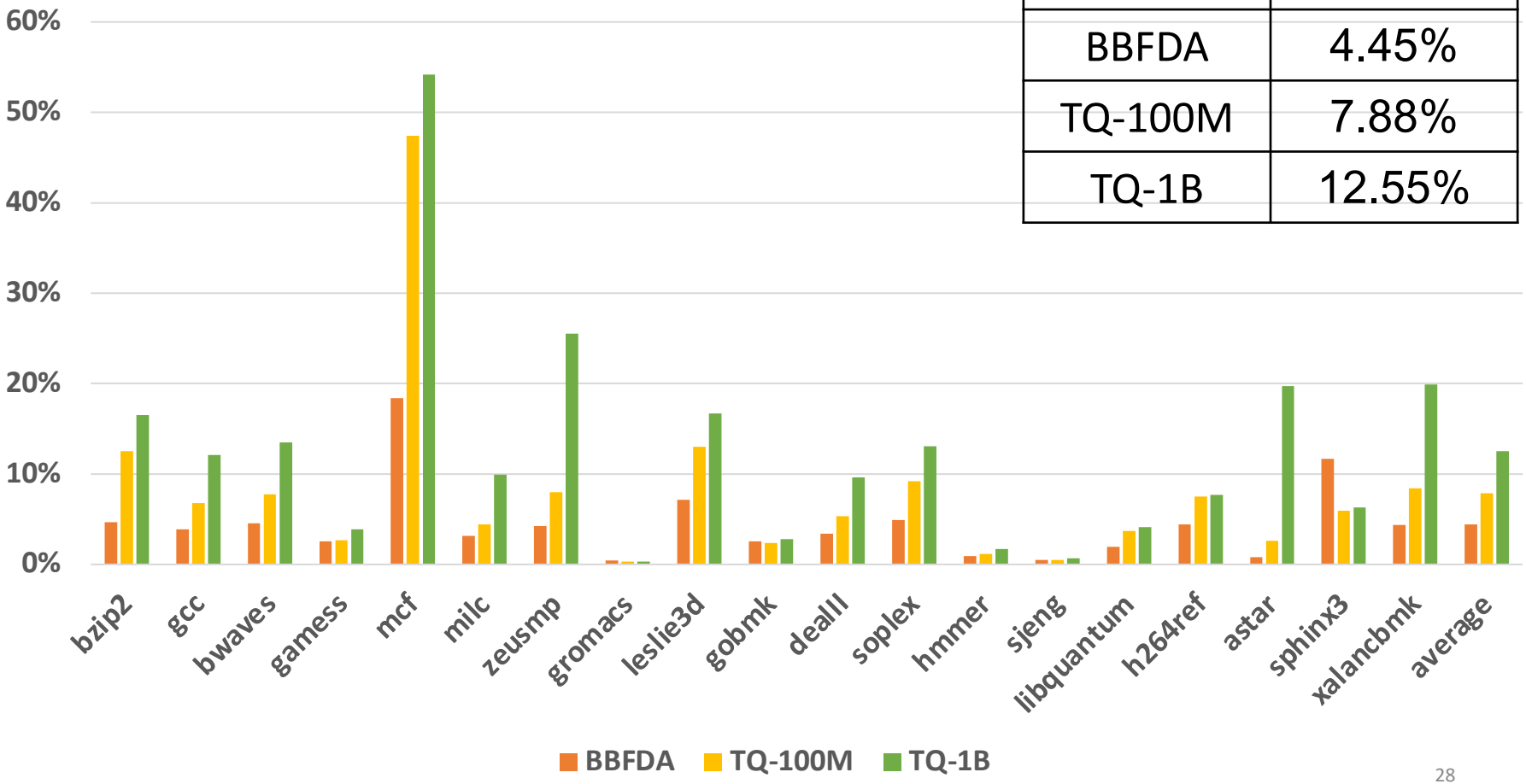
- Compute the weighted average from the performance value p_b of the basic block b that occurs n_b times in the phase.

$$p_p = (\sum_b n_b p_b) / (\sum_b n_b)$$

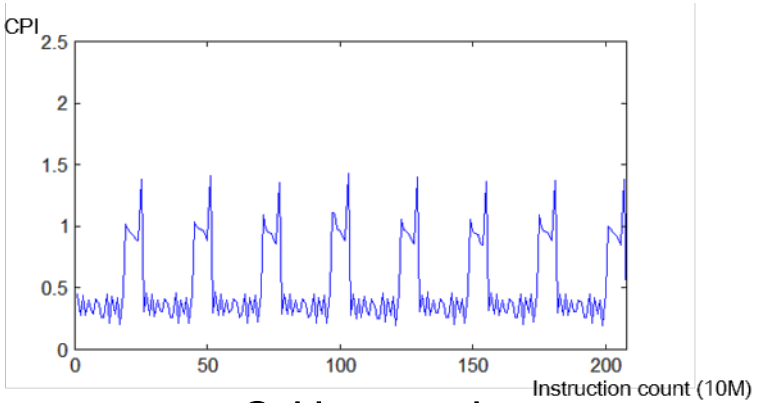


Estimation Error Rate

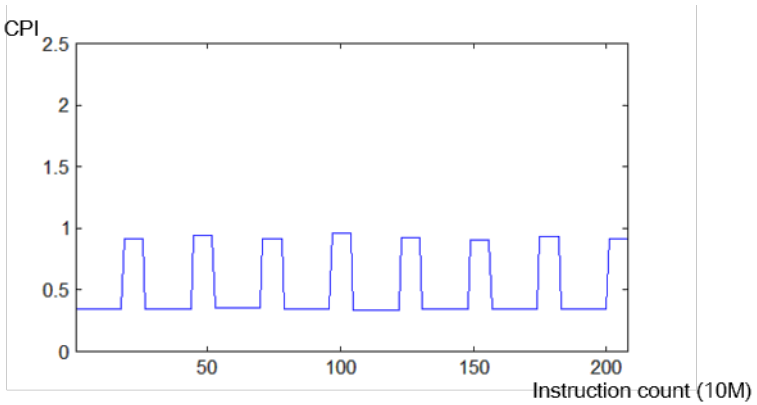
	average
BBFDA	4.45%
TQ-100M	7.88%
TQ-1B	12.55%



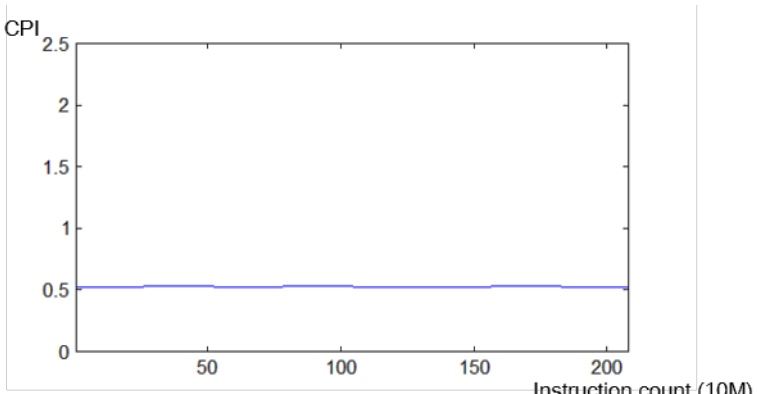
Compare Time Waveform Predictions



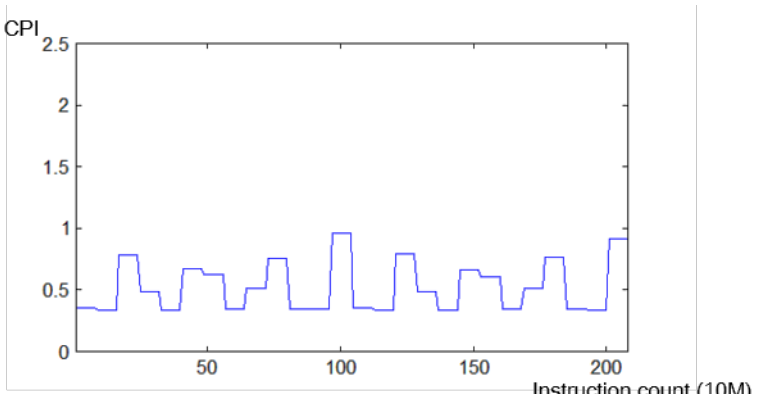
Golden sample



BBFDA

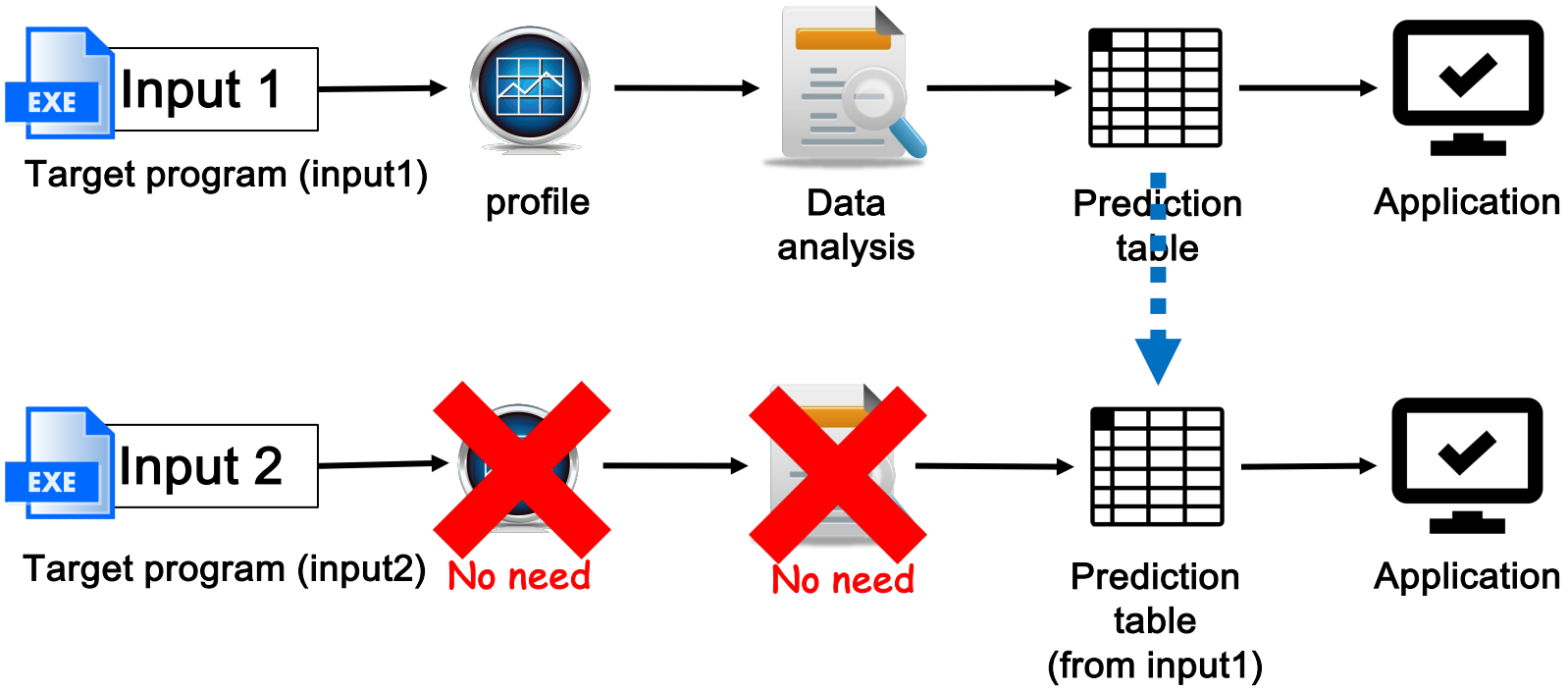


TQ-1B

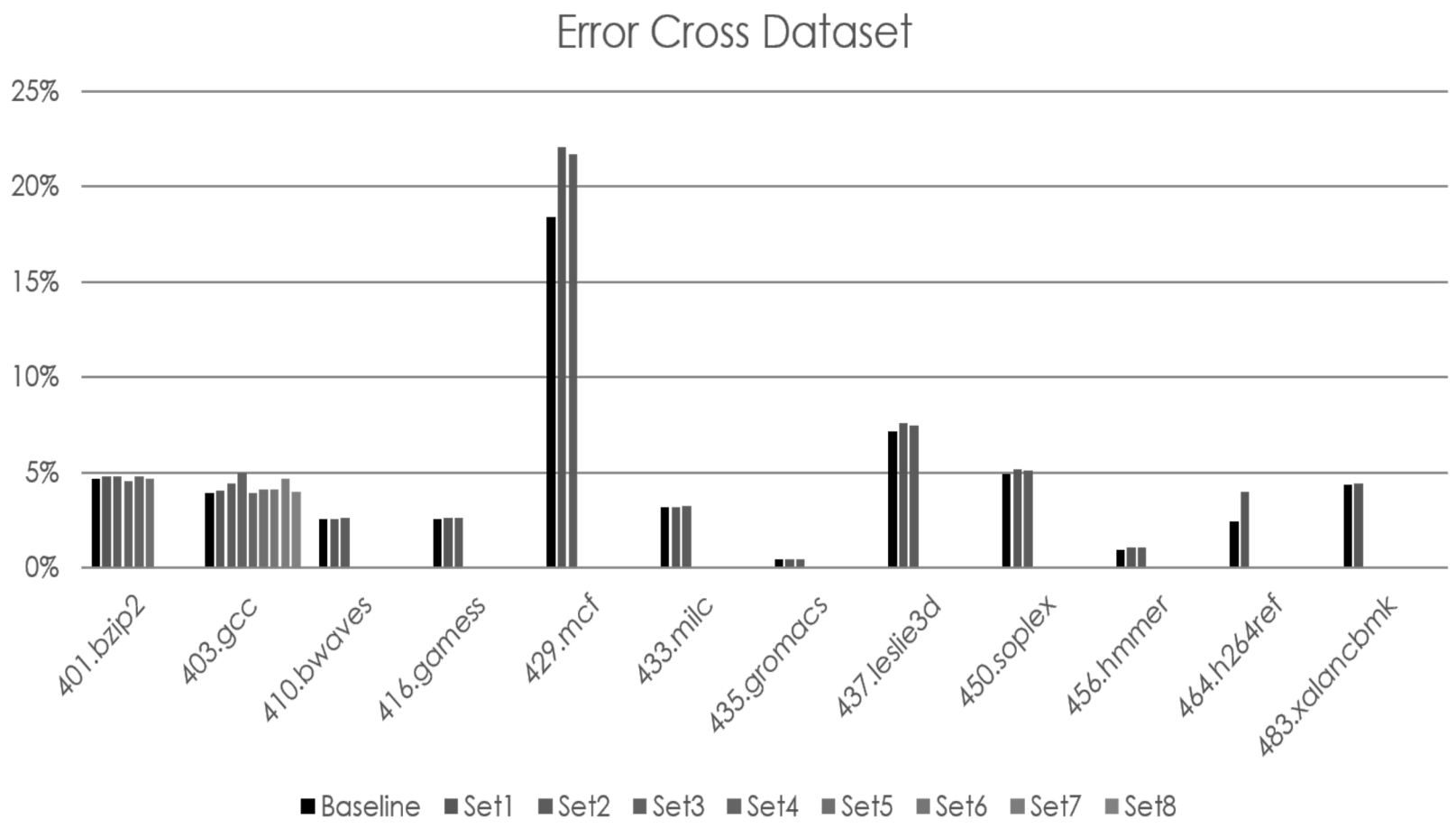


TQ-100M

Test Sensitivity to Input Data



Robust to Input Data



Conclusions

- **Efficient Program Phase Identification**: The BBFDA method effectively discerns program phases and precisely estimates performance waveforms.
- **Robust Input Handling**: Our approach remains resilient in the face of varying input data.
- **Time-Quantum Granularity Resolution**: Skillful avoidance of the time-quantum granularity issue.
- **Comprehensive Hierarchical Phase Information**: Detailed hierarchical phase information is supplied for optimal runtime adjustments.

Q & A