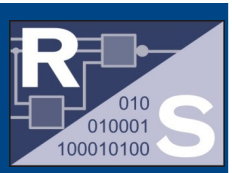
Hardware Support for Histogram-based Performance Analysis



Boris Dreyer
dreyer@rs.tu-darmstadt.de

Prof. Dr.-Ing. Christian Hochberger
Computer Systems Group
Department of Electrical Engineering and Information Technology
Technische Universität Darmstadt, Germany

This work was funded within the project CONIRAS by the German Federal Ministry for Education and Research with the funding ID 01IS13029. The responsibility for the content remains with the authors.

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Motivation**
  - Measurement-based Execution Time Estimation
- **Online Generated Basic Block Histograms**
- **Working with Basic Block Histograms**
  - Probability Mass Functions
  - Operations for PMFs
  - Example

Execution Time Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

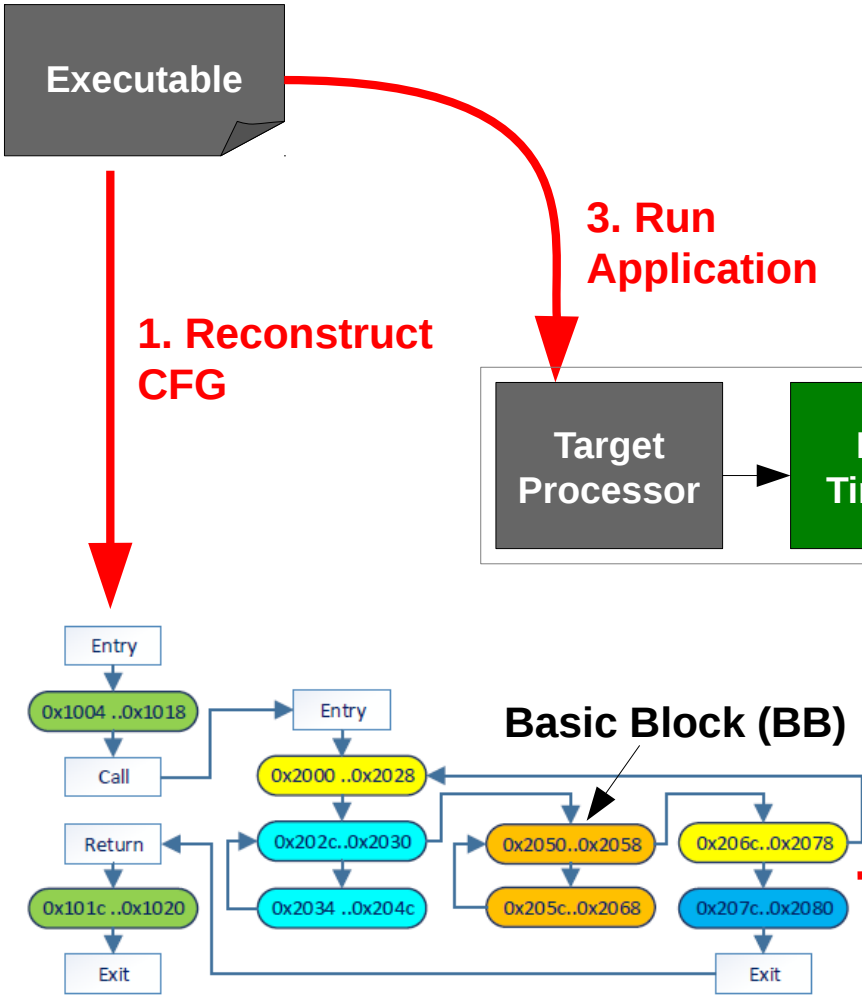
Boris Dreyer, Christian Hochberger, Simon Wegener, and Alexander Weiss.

Precise Continuous Non-Intrusive Measurement-Based Execution Time Estimation.

In Francisco J. Cazorla, editor, 15th International Workshop on Worst-Case Execution Time Analysis (WCET 2015), volume 47 of OpenAccess Series in Informatics (OASIcs), pages 45-54, Dagstuhl, Germany, 2015.

Schloss Dagstuhl—Leibniz-Zentrum für Informatik.

Execution Time Estimation - Idea



BB Address	Max First	Max Further
0x1004	8 us	0 us
0x2000	22 us	11 us
0x202c	16 us	8 us
0x2034	2 us	1 us
0x2050	12 us	8 us
0x205c	2 us	1 us
0x206c	8 us	4 us
0x207c	6 us	0 us
0x101c	6 us	0 us

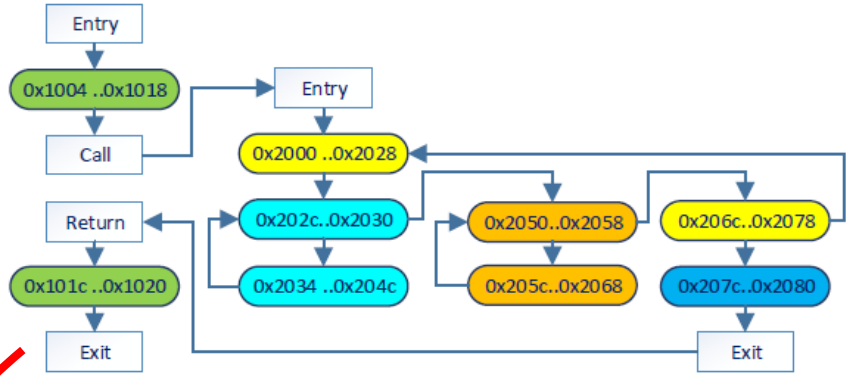
Context Sensitive BB Statistics

Execution Time Estimation - Idea

BB Address	Max First	Max Further
0x1004	8 us	0 us
0x2000	22 us	11 us
0x202c	16 us	8 us
0x2034	2 us	1 us
0x2050	12 us	8 us
0x205c	2 us	1 us
0x206c	8 us	4 us
0x207c	6 us	0 us
0x101c	6 us	0 us

Context Sensitive BB Statistics

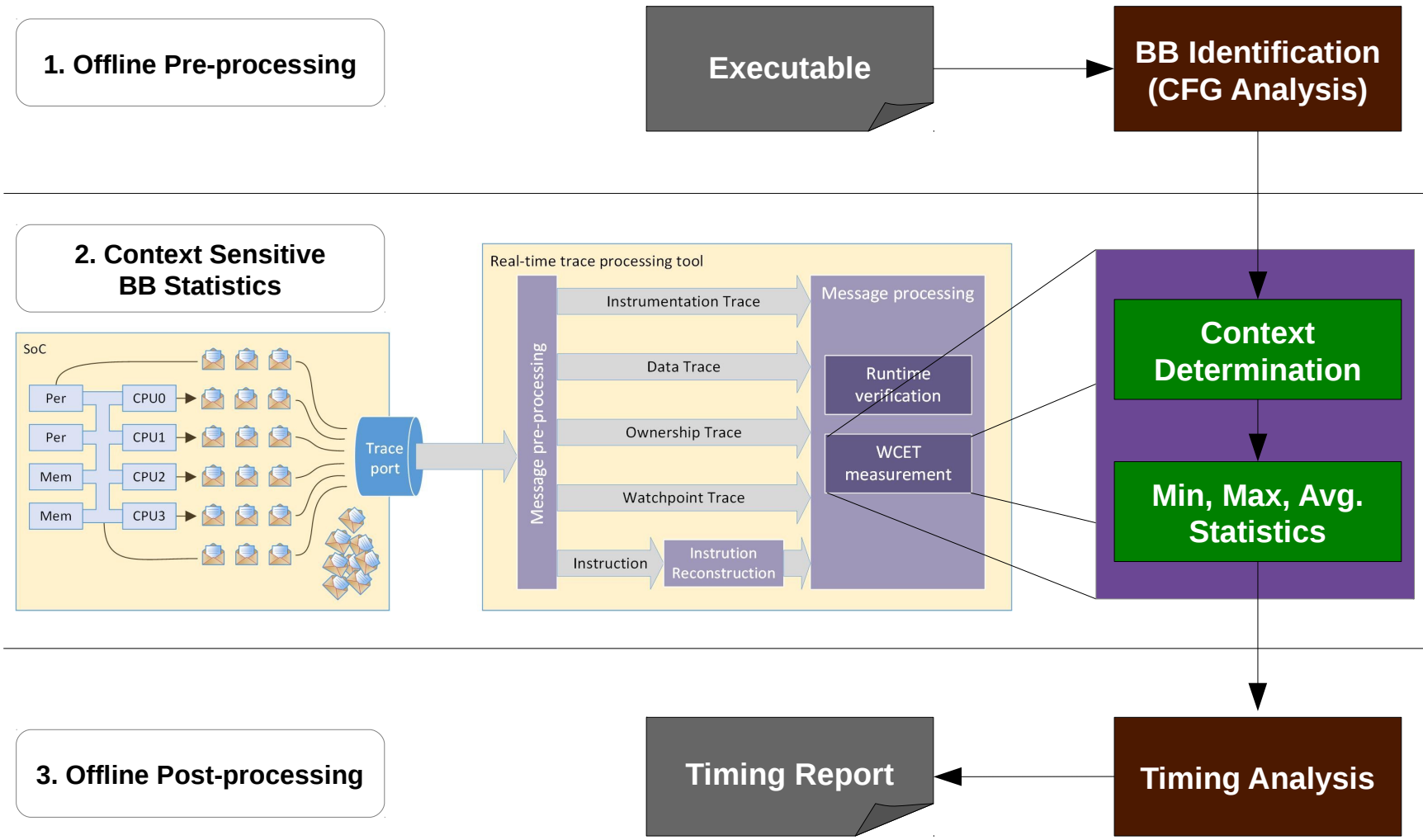
5. Annotate CFG



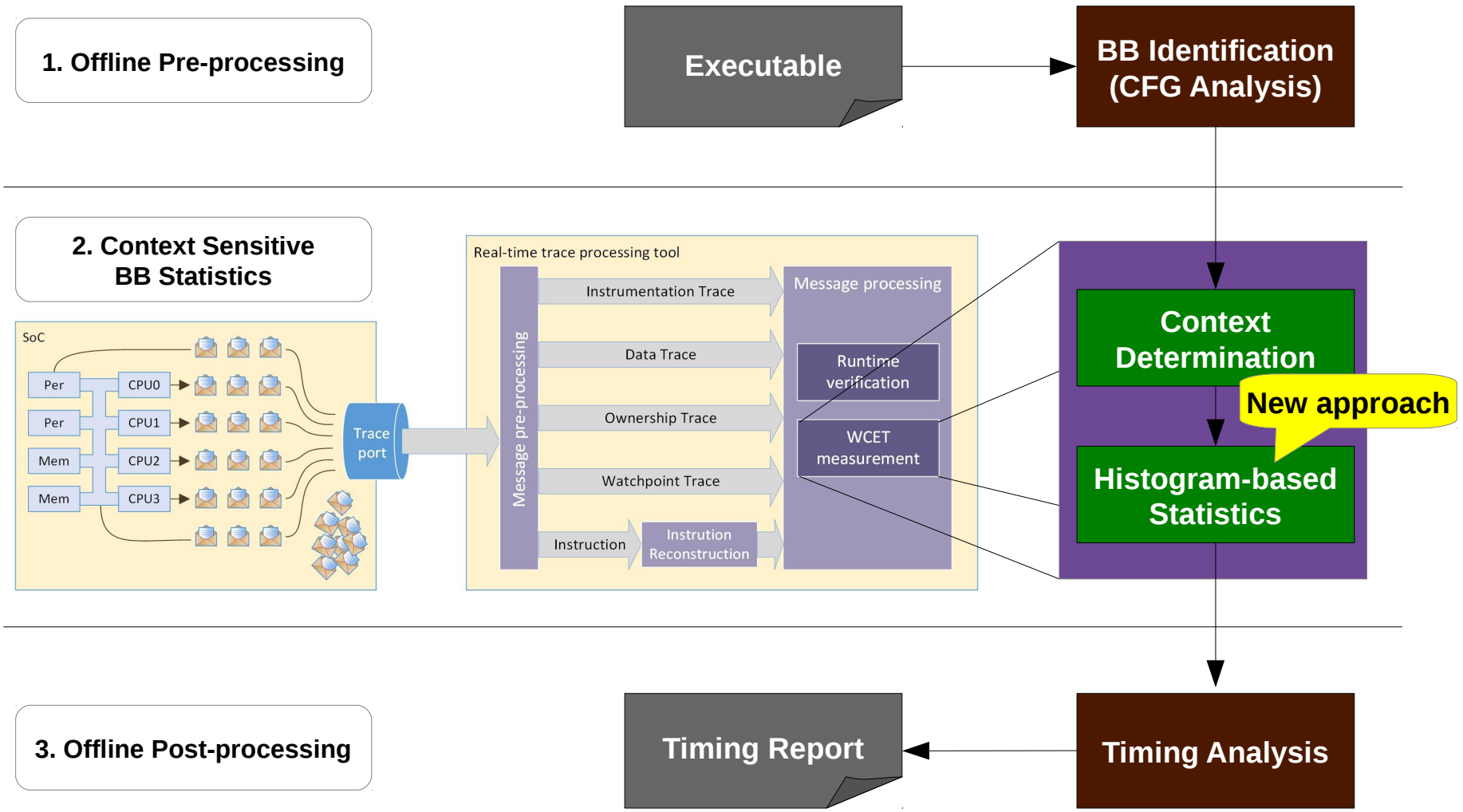
6. Find longest path (ILP based)

Overall execution time estimate
Our method: 191 us
Context insensitive: 258 us

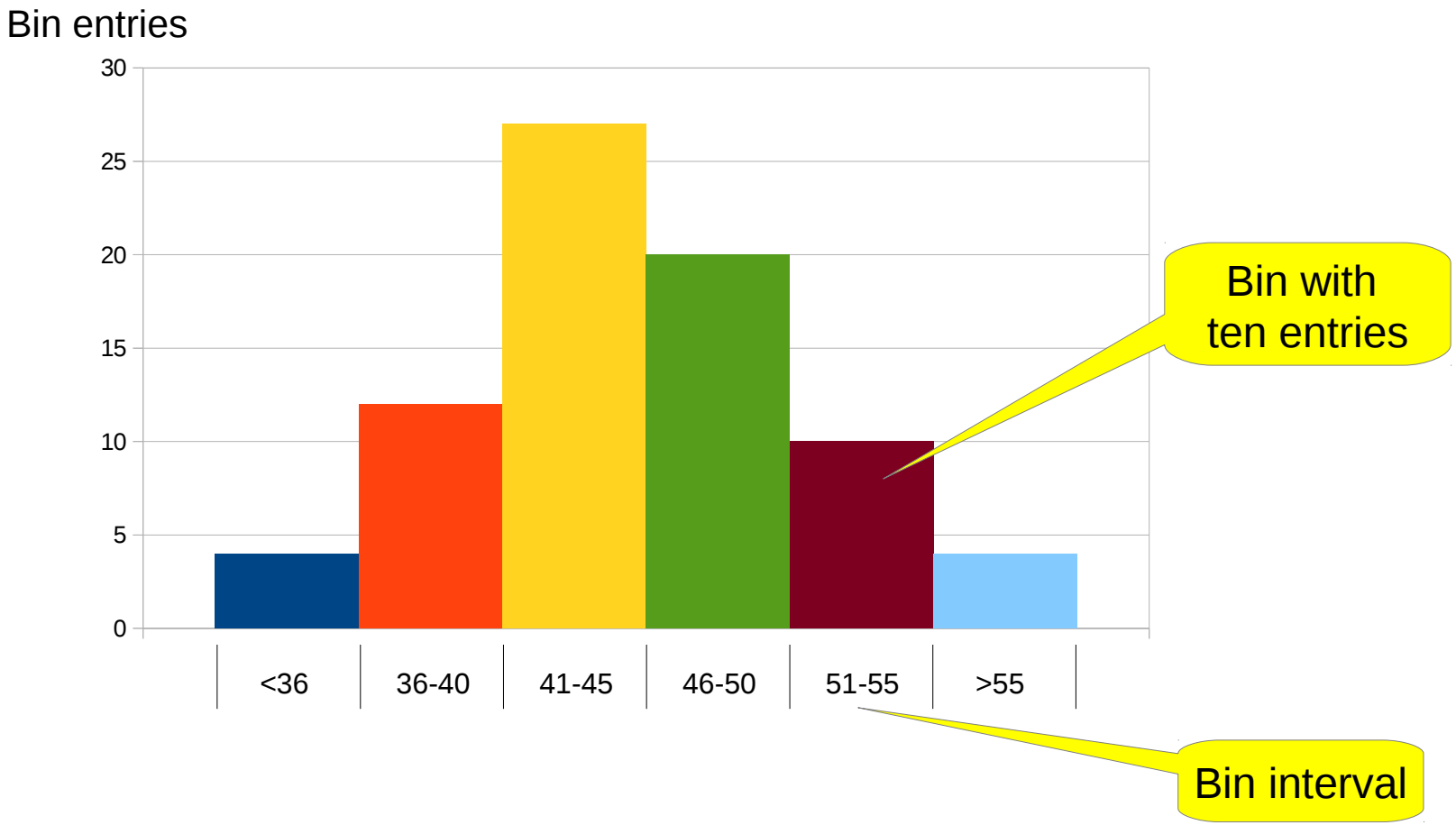
Execution Time Estimation - Architecture



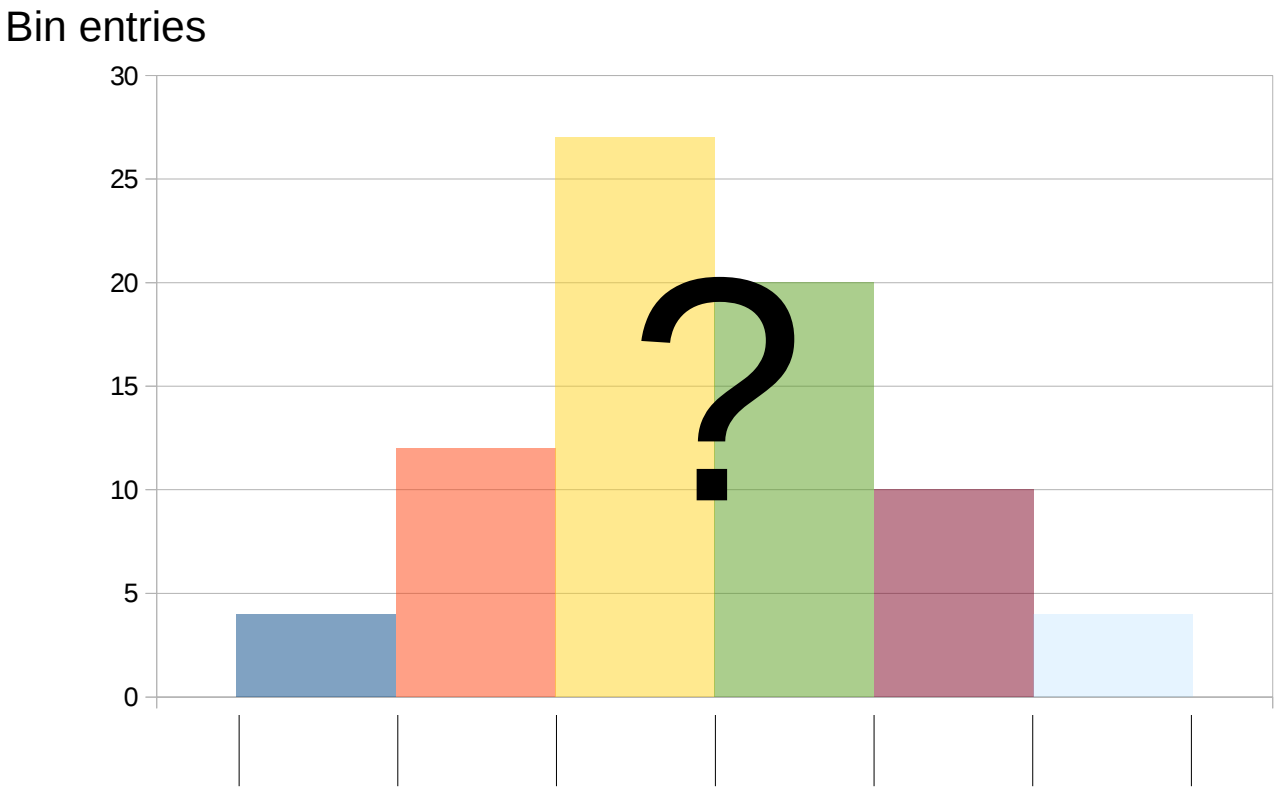
Execution Time Estimation - Architecture



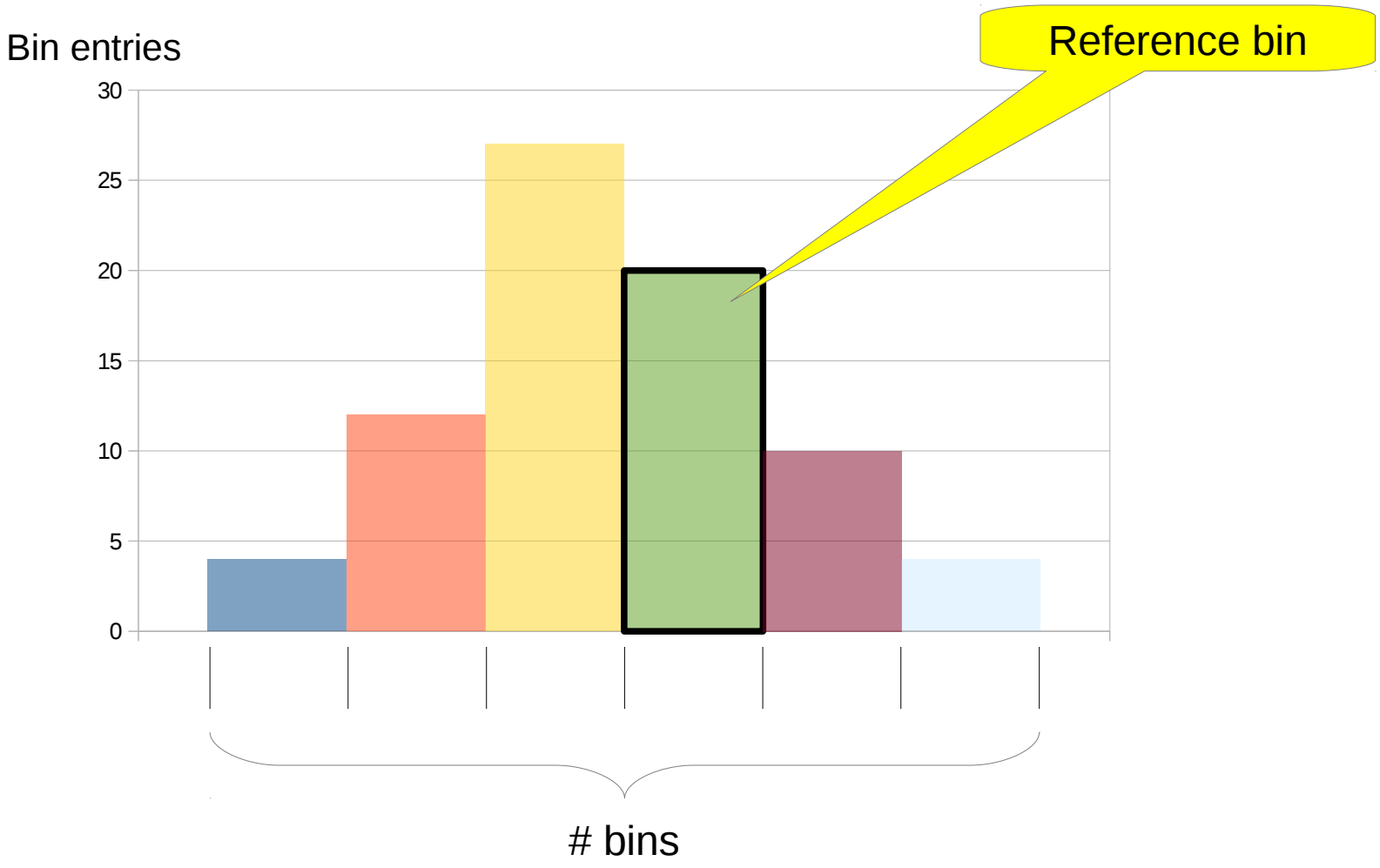
Histogram


Bin entries
30
25
20
15
10
5
0
<36
36-40
41-45
46-50
51-55
>55
Bin with ten entries
Bin interval

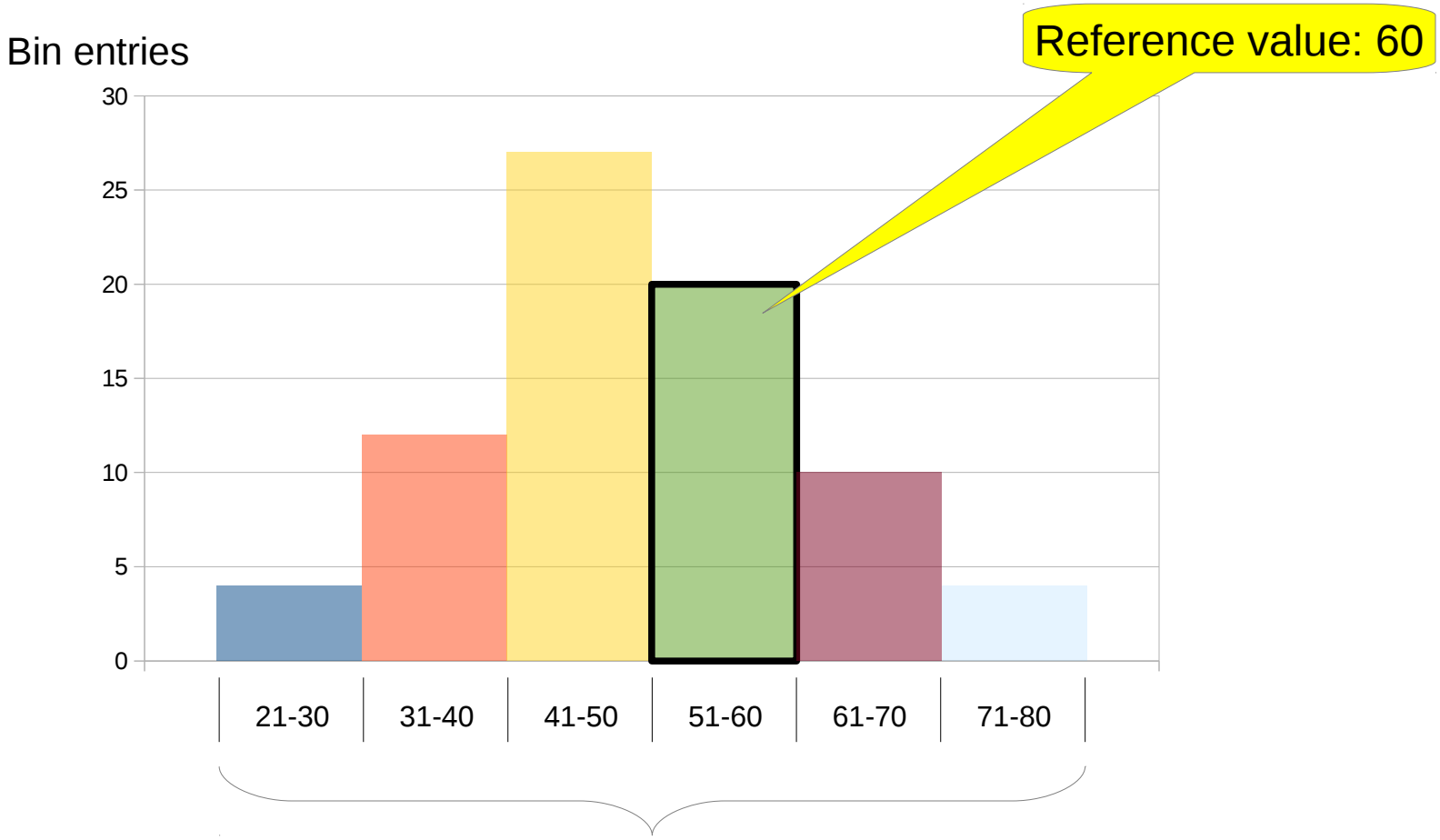
Histogram – Implementation



Histogram – Reference Bin

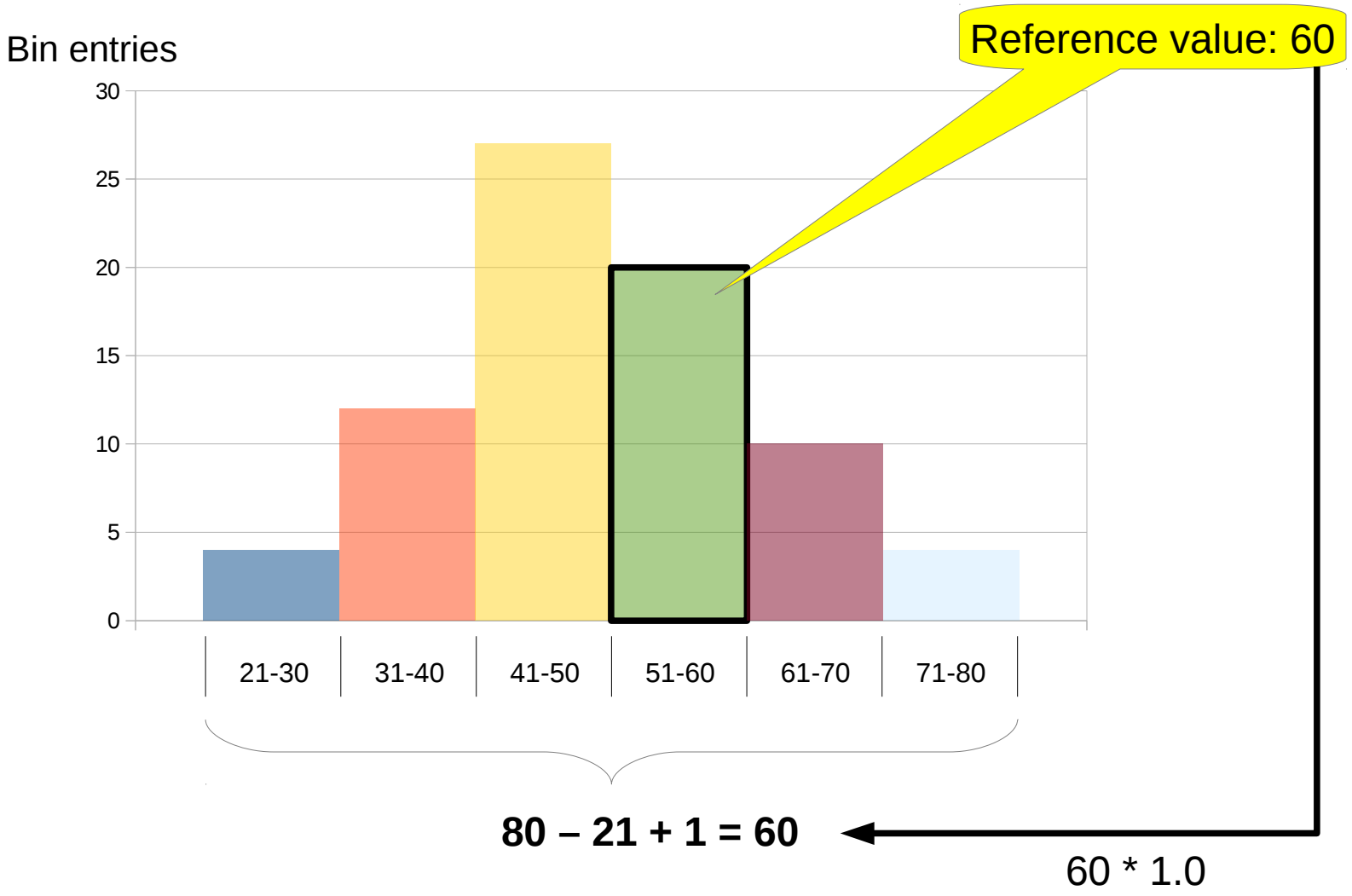


Histogram – Reference Value

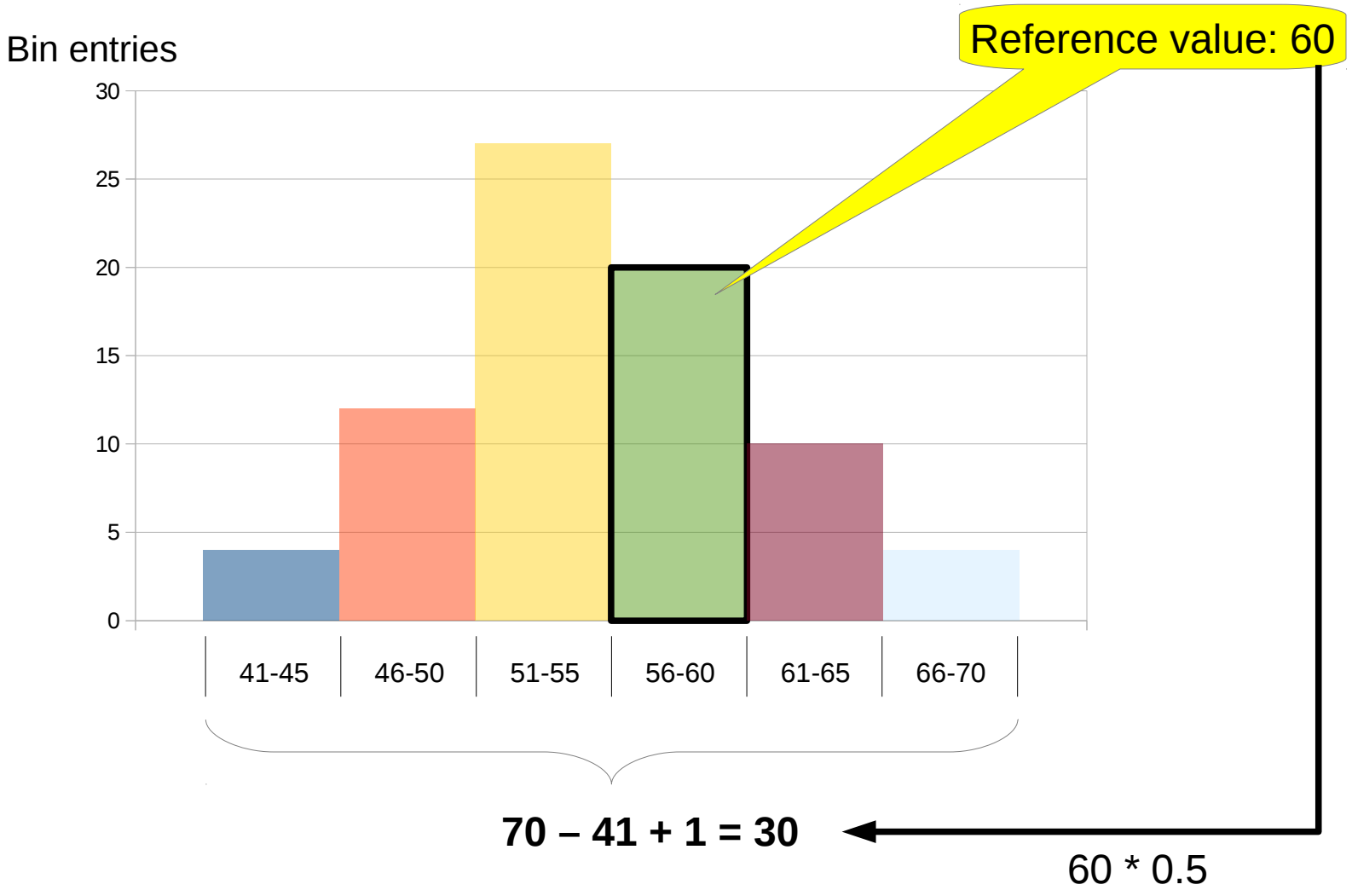


80 – 21 + 1 = 60

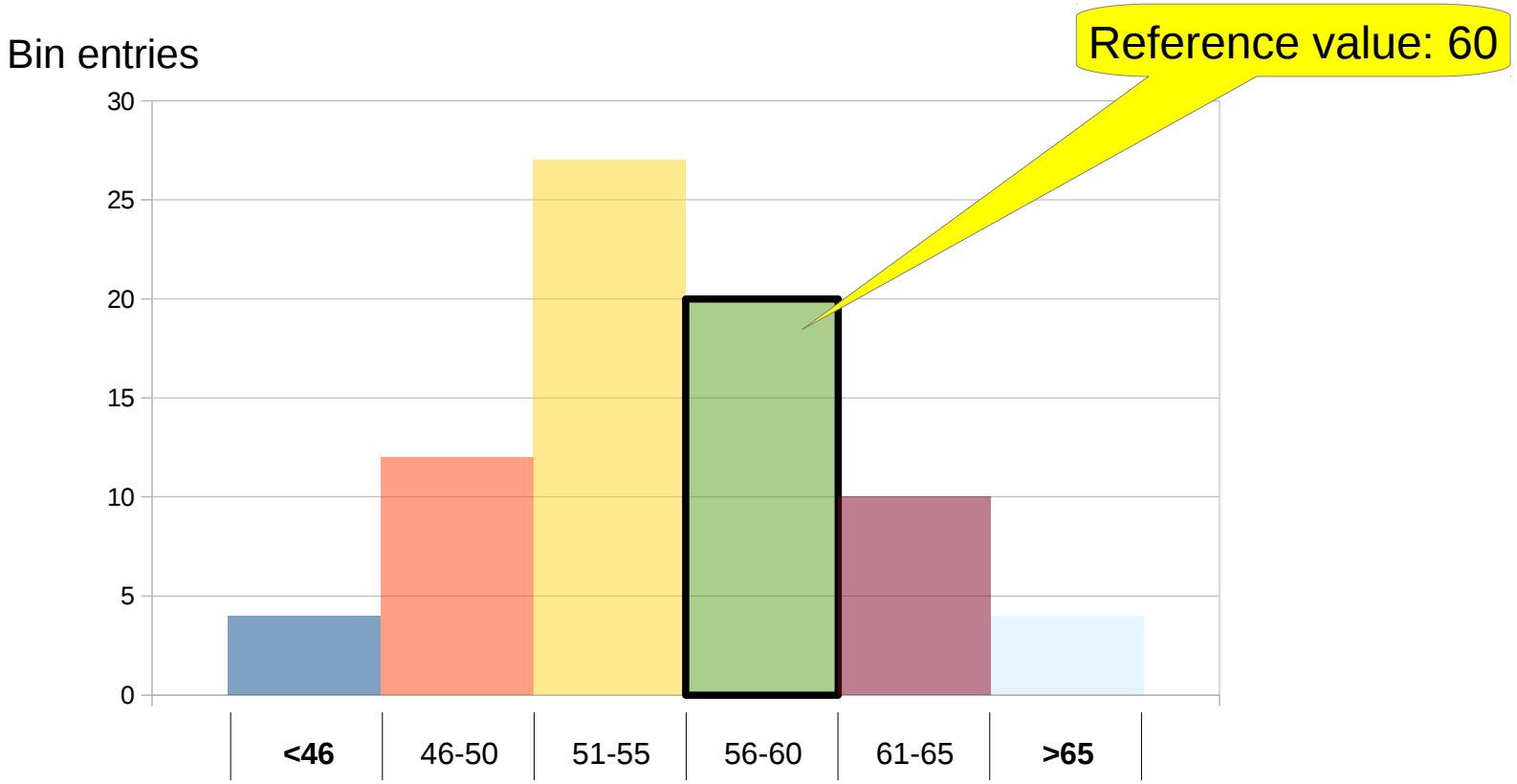
Histogram – Linear Compression = 1.0



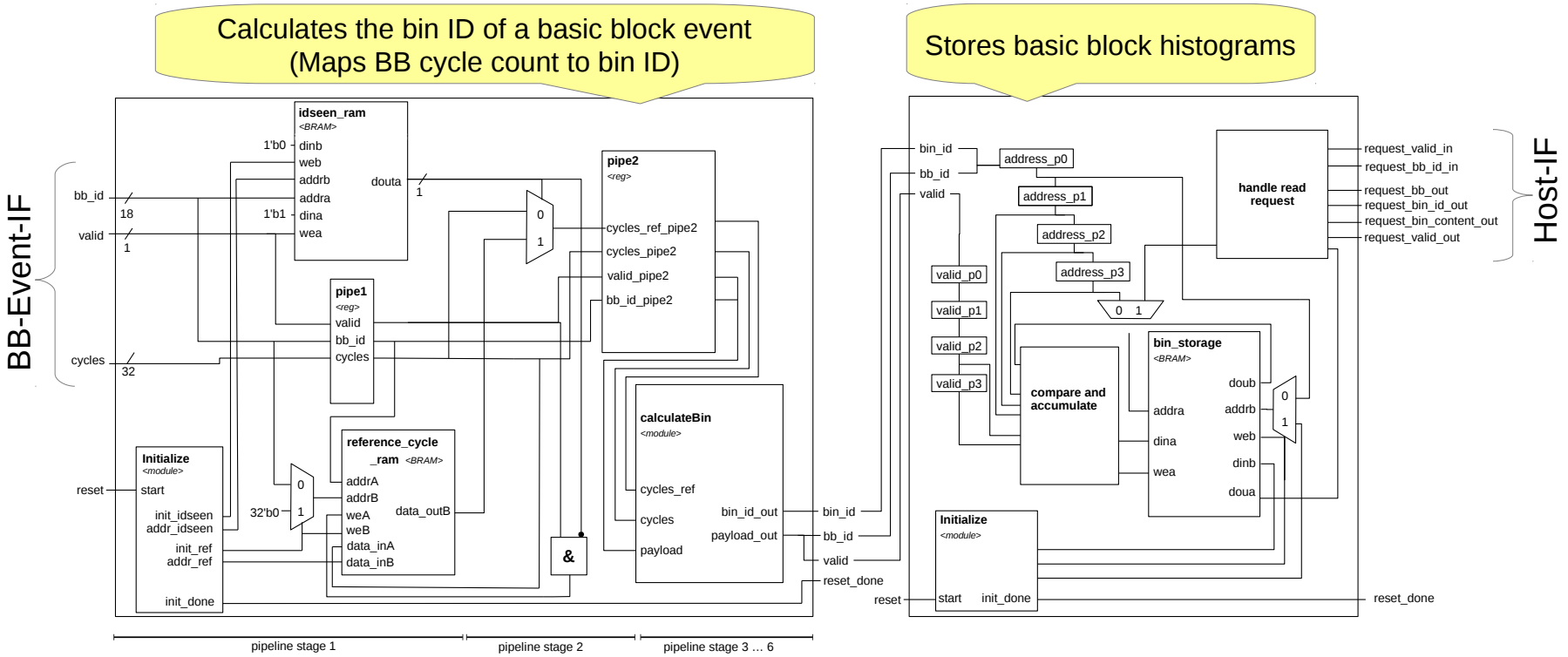
Histogram – Linear Compression = 0.5



Histogram – Catch-all Bins



Histogram Implementation



Bin finder module parameter

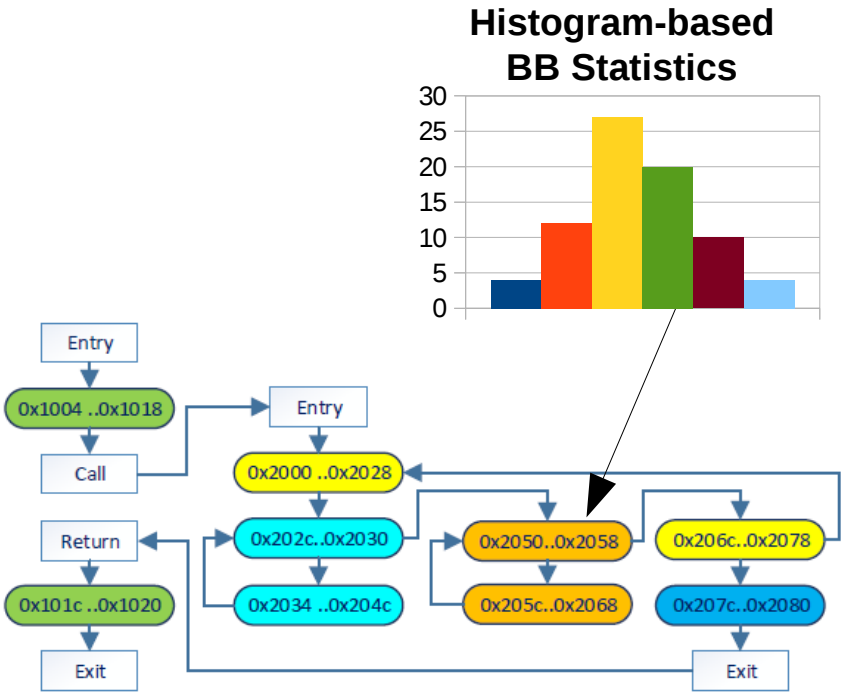
- # basic blocks
- # bins per basic block
- reference bin
- linear compression

Storage module parameter

- # basic blocks
- # bins per basic block

Determine resource consumption

Histogram-based BB Statistics



Histograms at Different Abstraction Levels