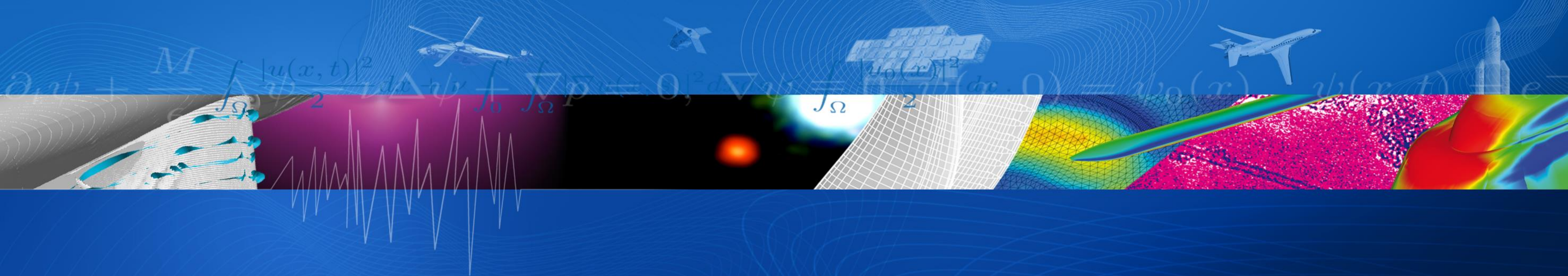


MULTICORE SHARED MEMORY INTERFERENCE ANALYSIS THROUGH HARDWARE PERFORMANCE COUNTERS

Alfonso Mascareñas González
Youcef Bouchebaba
Luca Santinelli

10th EDITION
EUROPEAN CONGRESS
ERTS 2020
FROM 29th TO 31st JANUARY / TOULOUSE FRANCE
PIERRE BAUDIS CONGRESS CENTER
EMBEDDED REAL TIME SYSTEMS
3AF
see

ONERA

THE FRENCH AEROSPACE LAB

PLAN

OBJECTIVES

- Design and validate a Performance Monitor Hardware measurement based framework
- Analyze memory interference within a multicore system
- Check the pWCET applicability on the obtained results

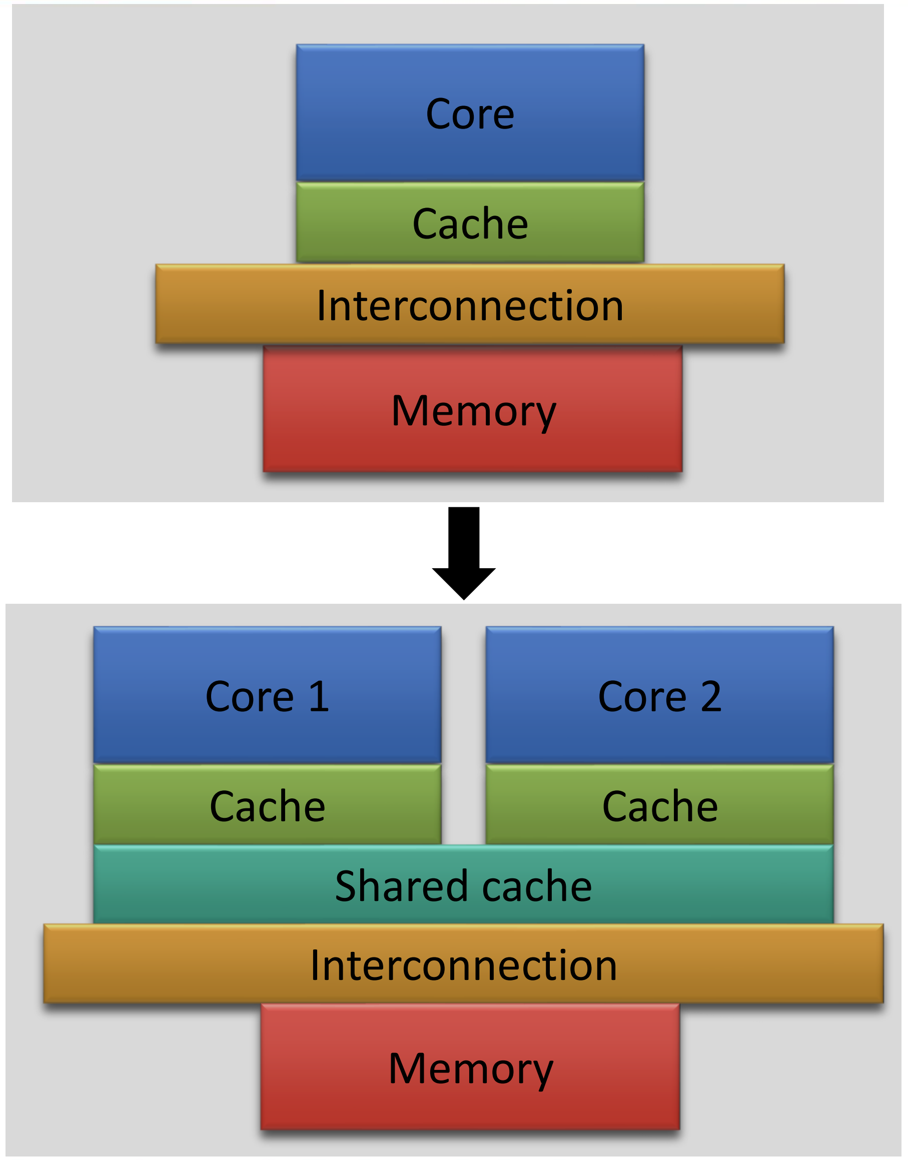
BACKGROUND

- Critical application: Meet timing conditions
- Single core vs Multicore processor systems
- Multicore systems

  + Throughput

  + SWaP (Size, Weight and Power)

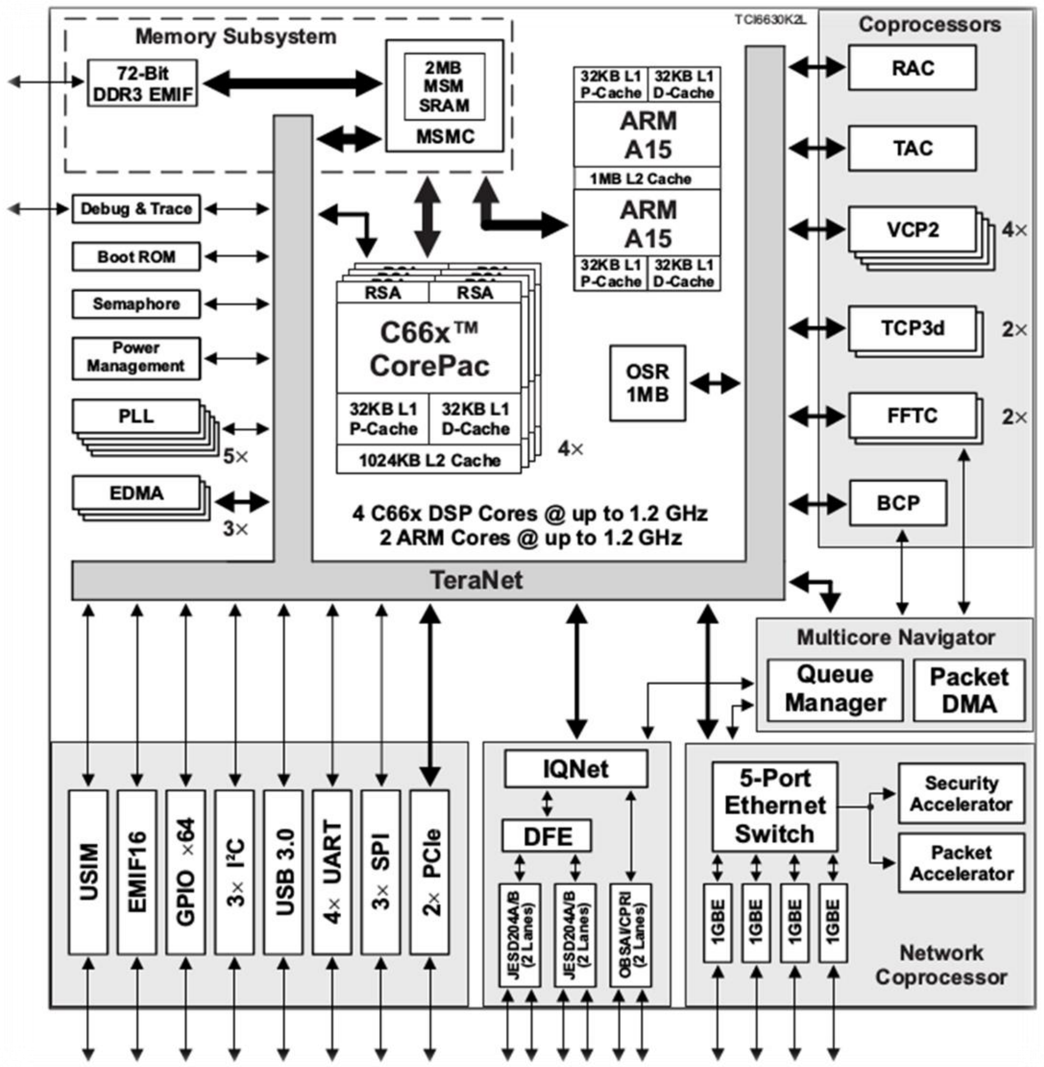
  - Predictability: Interference within the whole platform increases
- Timing analysis: Tasks Worst Case Execution Time (WCET) to Tasks probabilistic WCET (pWCET)



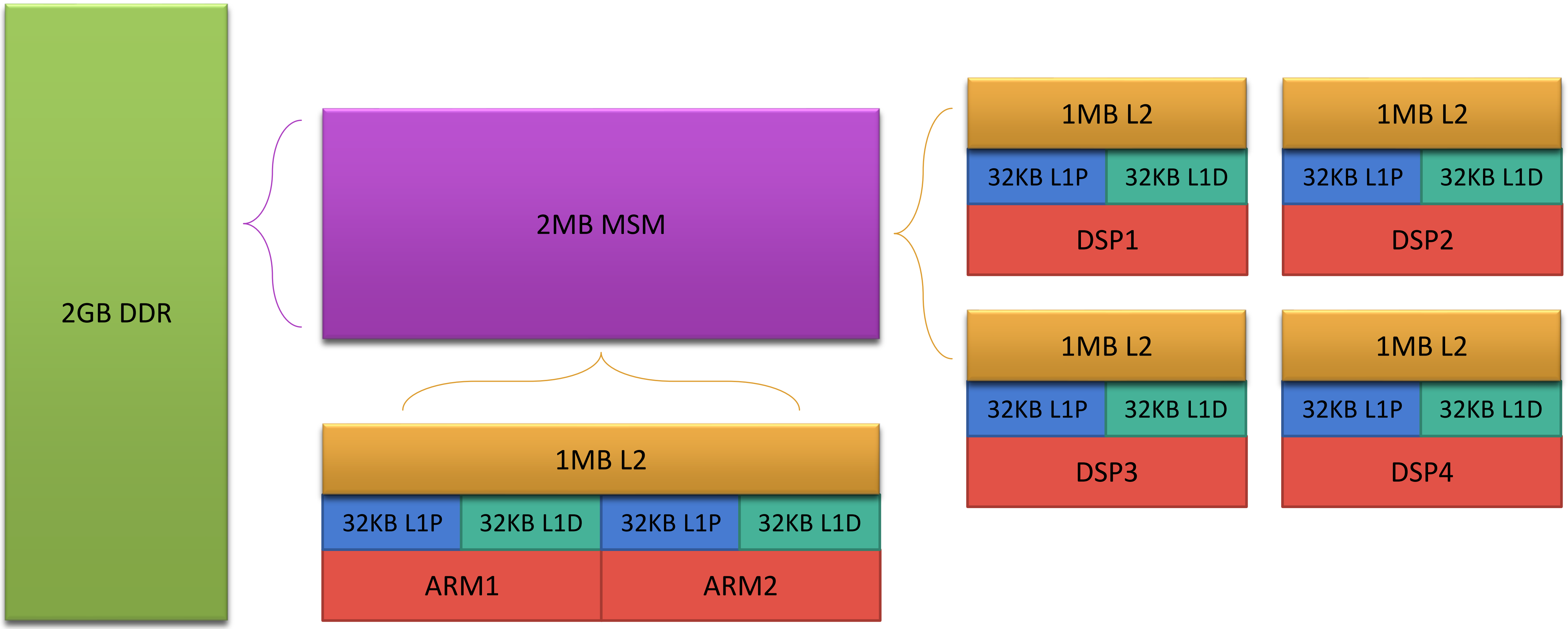
MULTICORE DEVICE: OVERVIEW

Keystone II TCI6630K2L

- 2 ARM cores @ 1.2GHz
- 4 DSP cores @ 1.2GHz
- L1, L2 cache memories
- MSM SRAM and DDR3 memories



MULTICORE DEVICE: MEMORY ORGANIZATION



MEASUREMENT FRAMEWORK

- Performance Monitor Hardware (PMH):
  - Coprocessors
  - Performance Monitor Unit (PMU): 6 general counters + 1 cycle specific counter
- Start-read access pattern:
  1. Selection of the counter
  2. Selection of the event
  3. Enable counter
  4. Reset counter
  5. Read actual counter value (first time)
  6. Run critical task
  7. Read actual counter value (second time) and make the difference

Events (~ 80)
L1 data cache refill
L1 data cache access
Mispredicted branch speculatively executed
Execution cycles
L2 data cache access
L2 data cache refill
L2 data cache Write-Back
Bus access
Data memory access
…

TASKS DESIGN

- The real-time applications:
  - Critical task: The one under observation. Three stressing levels to choose (safety1, safety2, safety3)
  - Non-critical tasks: Act as memory stressing source

Loops
Simple operations
Matrices: Main memory demanding source

Tasks are continuously being executed. They are structured as follows:

- Critical task in 1 ARM
- Non-critical task in 1 ARM and 4 DSPs

ARMs are managed by PikeOS

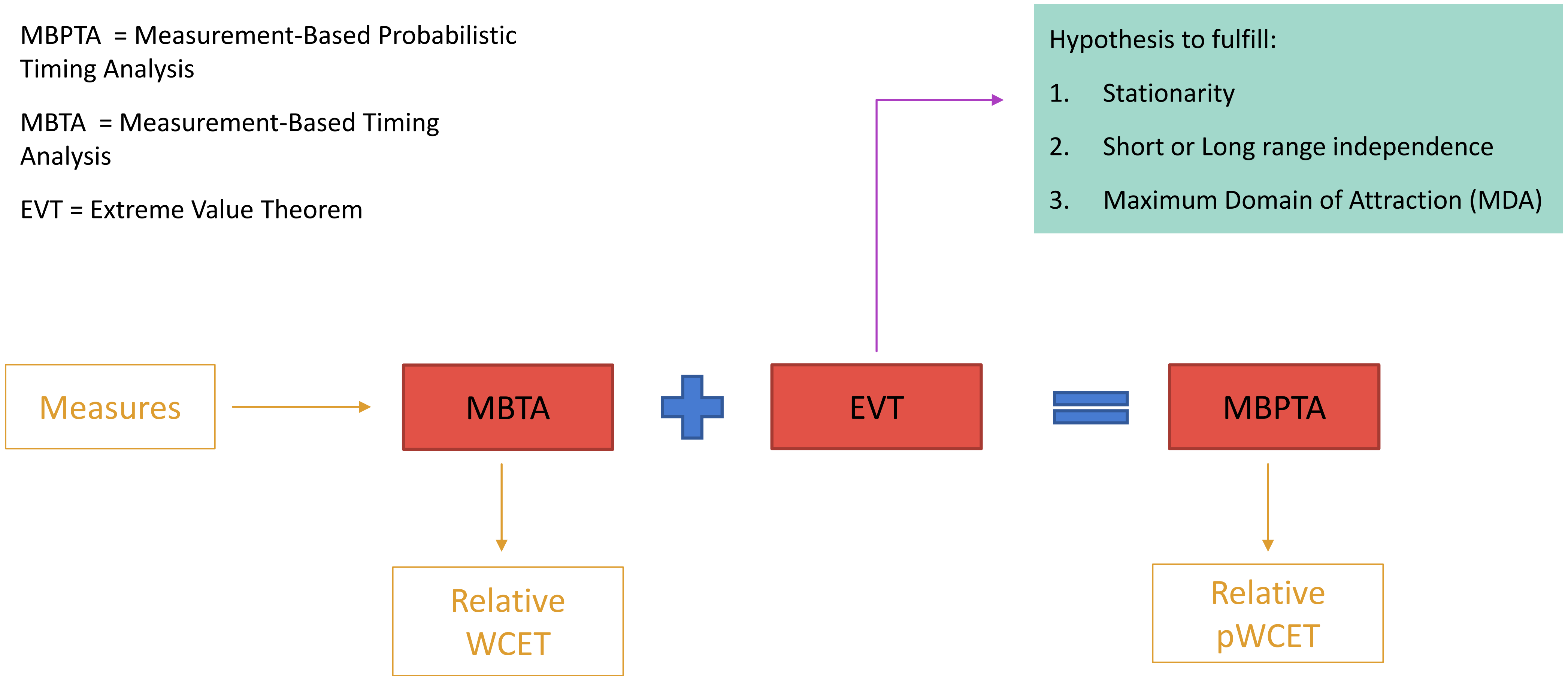
DSPs are fully bare metal

STATISTICAL APPLICATION: pWCET & EVT

MBPTA = Measurement-Based Probabilistic Timing Analysis

MBTA = Measurement-Based Timing Analysis

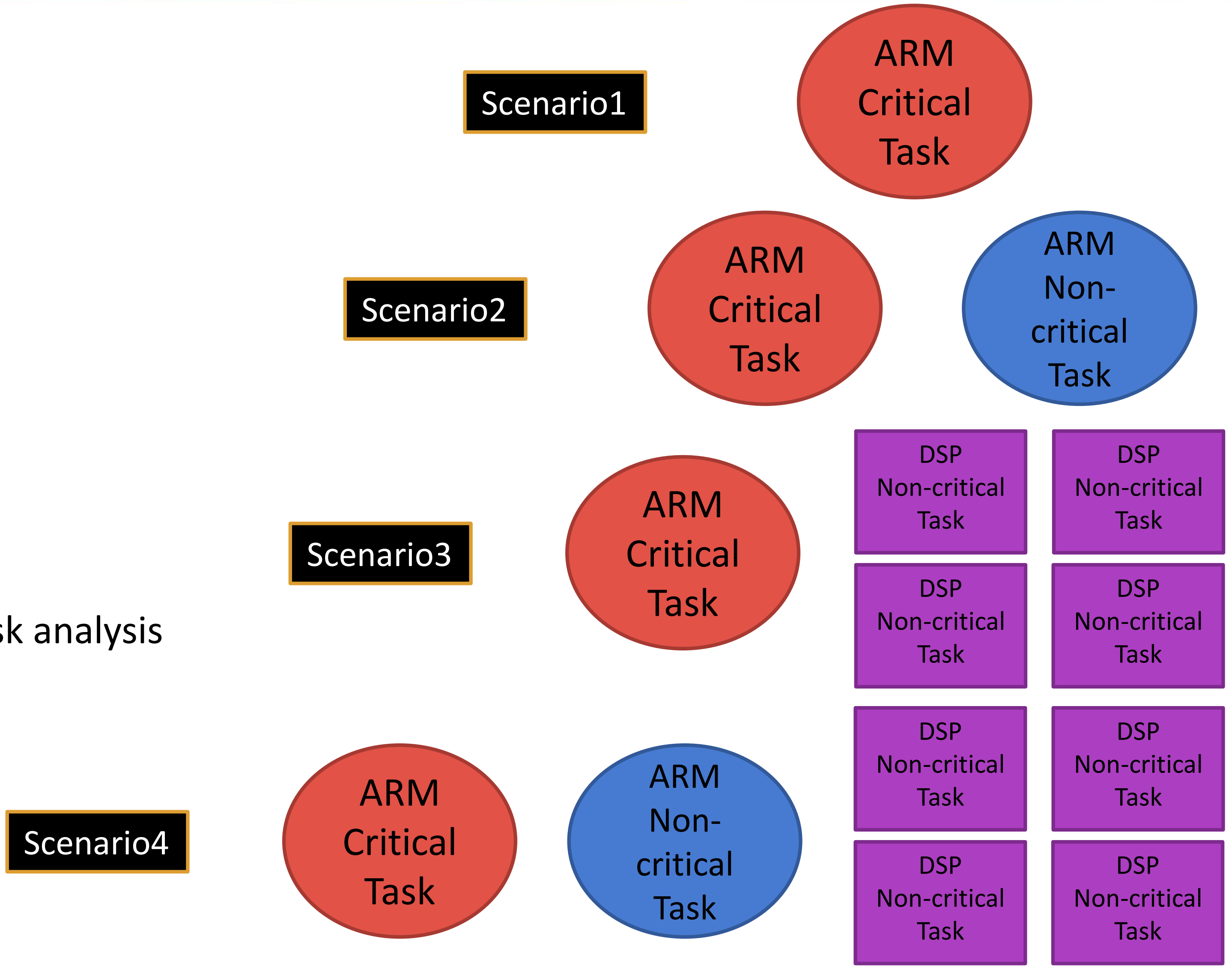
EVT = Extreme Value Theorem



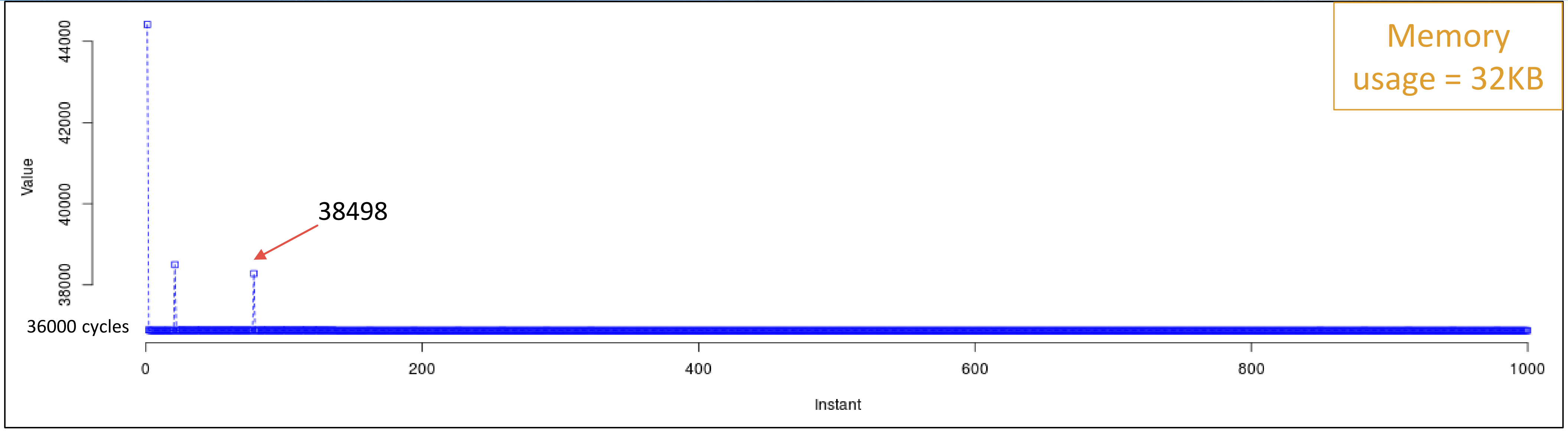
SCENARIOS: DESIGN

Four possible scenarios:

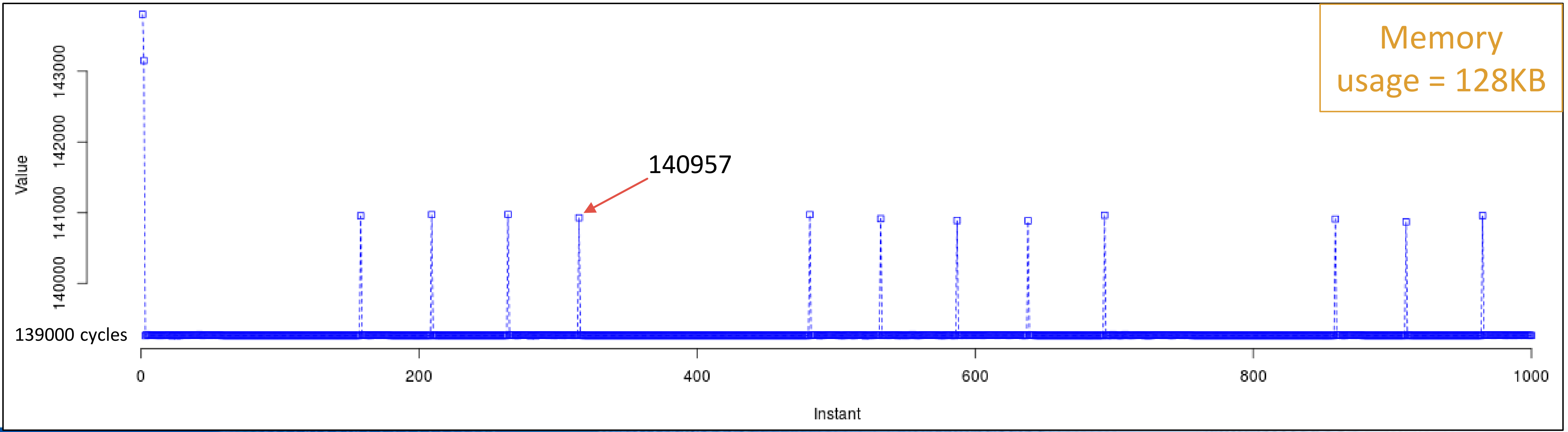
1. Critical task analysis
2. Critical task + ARM non-critical task analysis
3. Critical task + DSPs non-critical task analysis
4. Critical task + ARM and DSPs non-critical task analysis



SCENARIO 1 RESULTS: EXECUTION CYCLES (SAFETY1)

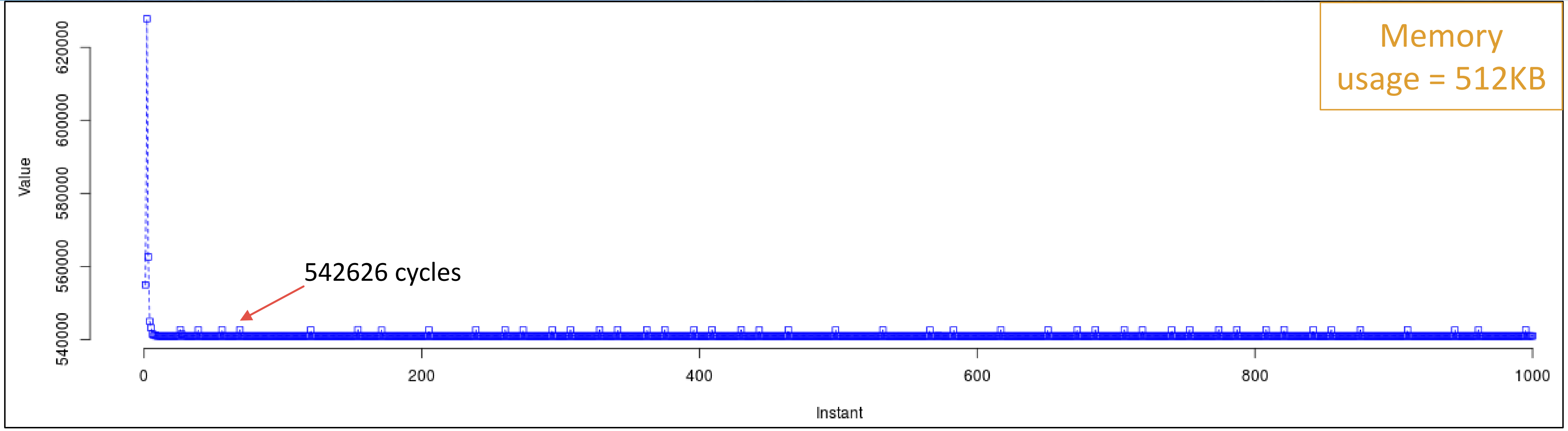


L1-L2



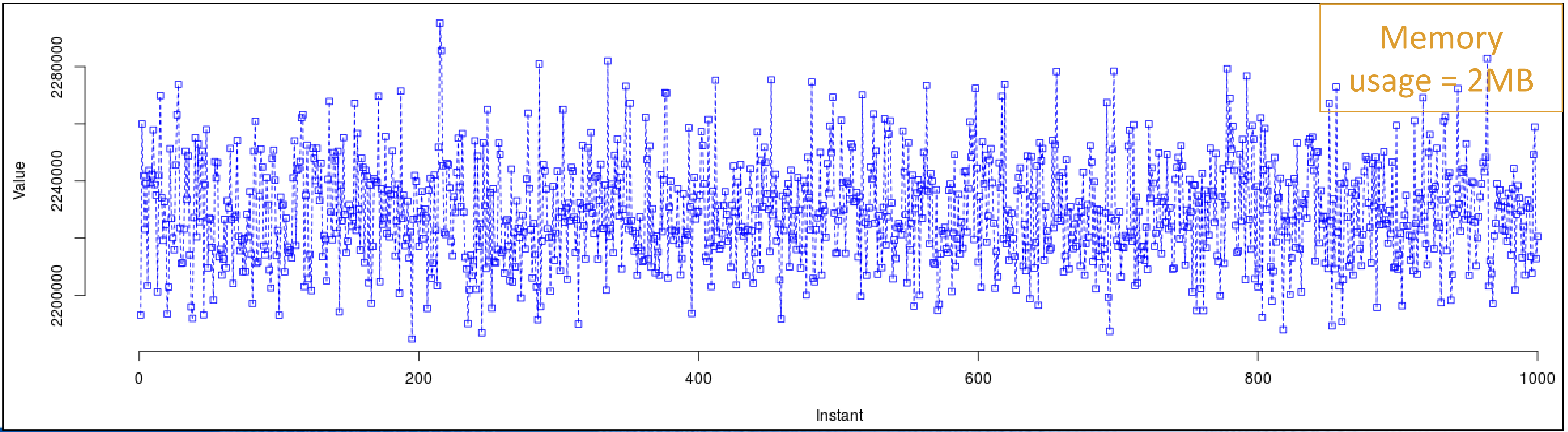
L2

SCENARIO 1 RESULTS: EXECUTION CYCLES (SAFETY1)



Memory usage = 512KB

L2

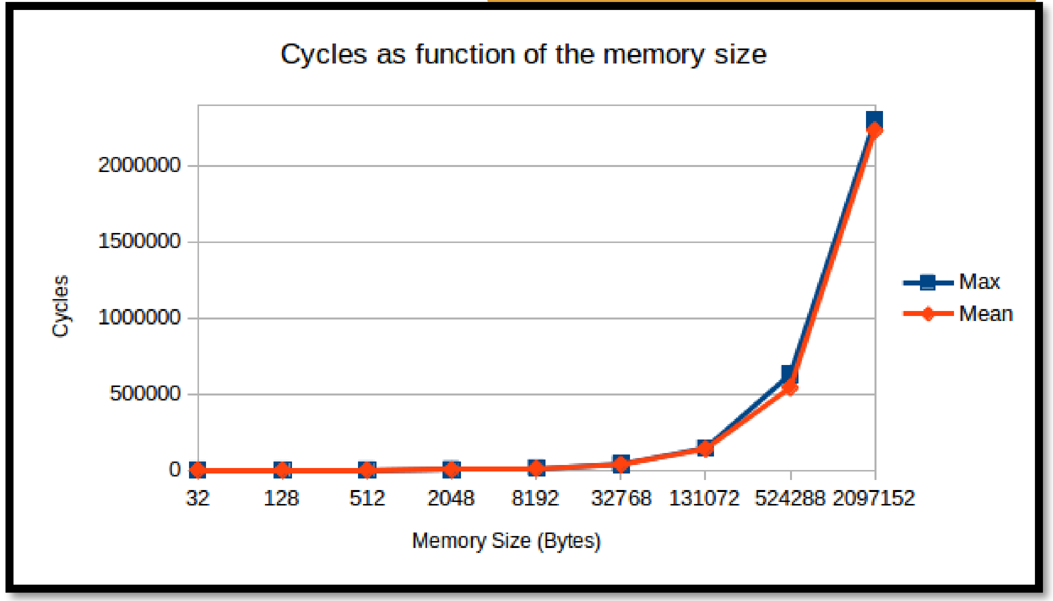


Memory usage = 2MB

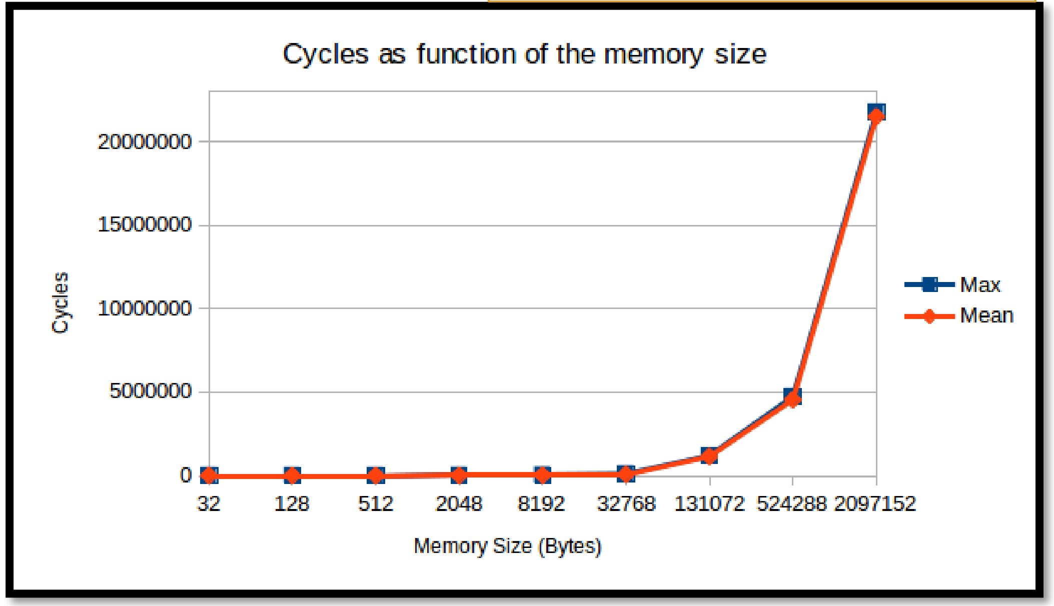
DDR

SCENARIO 1 SUMMARY: EXECUTION CYCLES

Safety1

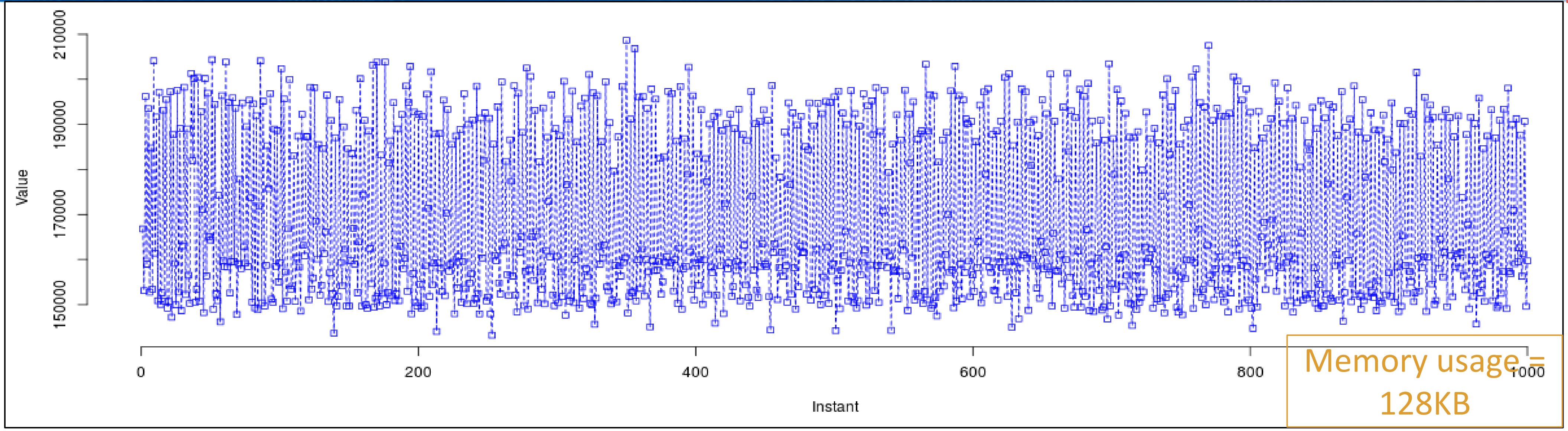


Safety3



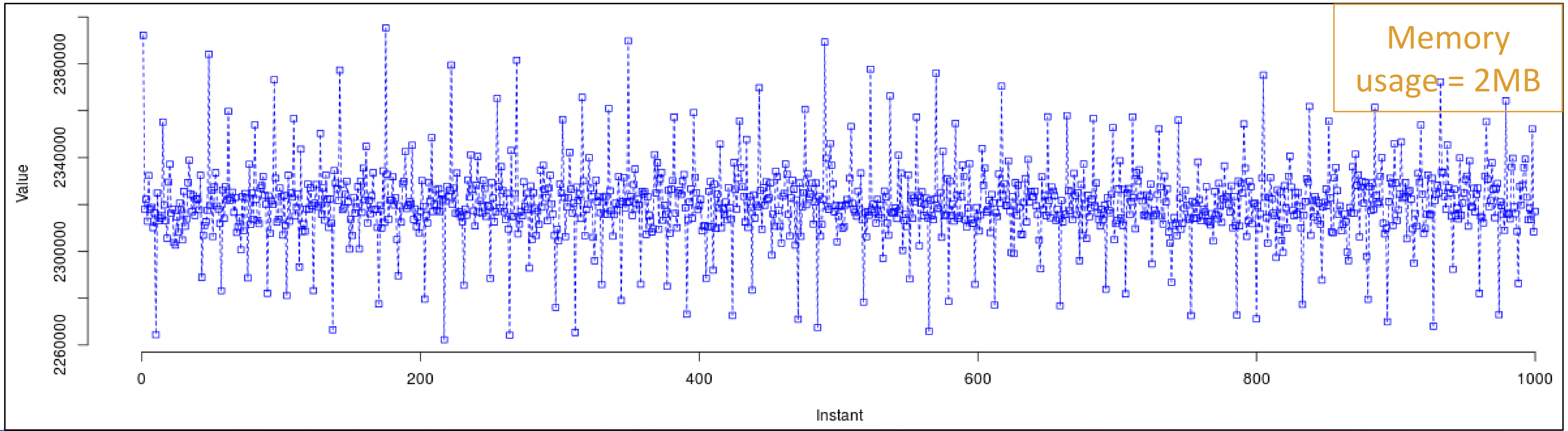
SCENARIO 2 RESULTS: EXECUTION CYCLES (SAFETY1)

Non-critical task memory usage = 2MB



Memory usage = 128KB

DDR



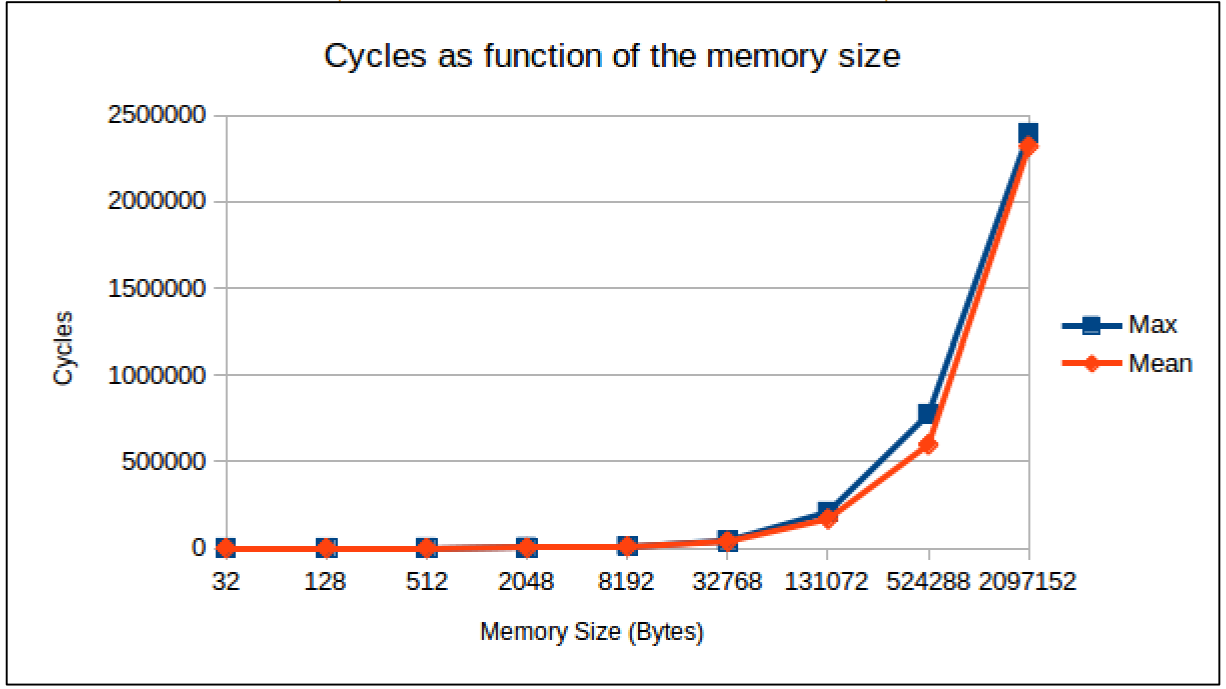
Memory usage = 2MB

DDR

SCENARIO 2 SUMMARY: EXECUTION CYCLES

Non-critical task memory usage = 2MB

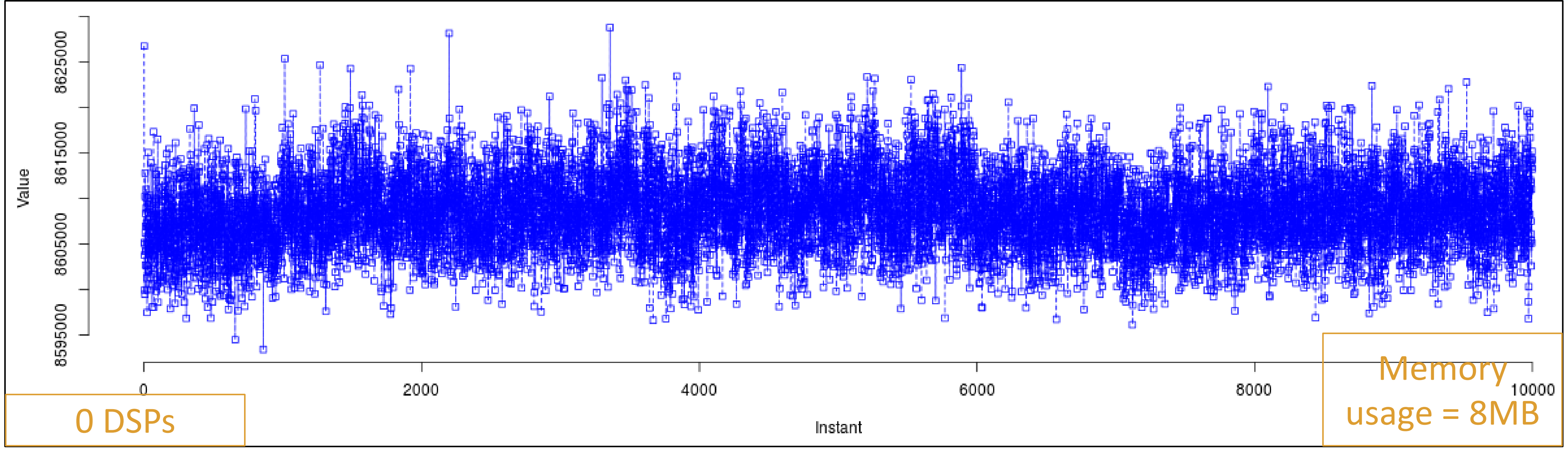
Safety1



Memory Size (KB)	Mean Overhead (%)	Max Overhead (%)
8	0,185	11,241
32	7,362	13,735
128	21,228	45,112
512	10,72	23,481
2048	4,091	4,363

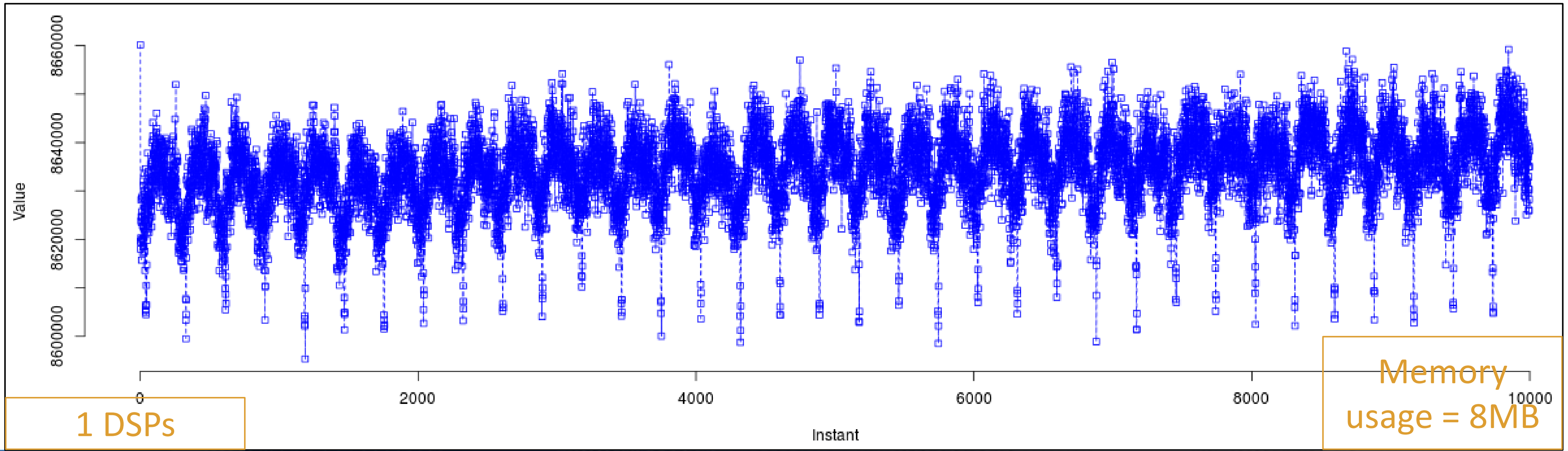
SCENARIO 3 RESULTS: EXECUTION CYCLES (SAFETY1)

Non-critical task memory usage = 12MB



0 DSPs

Memory usage = 8MB

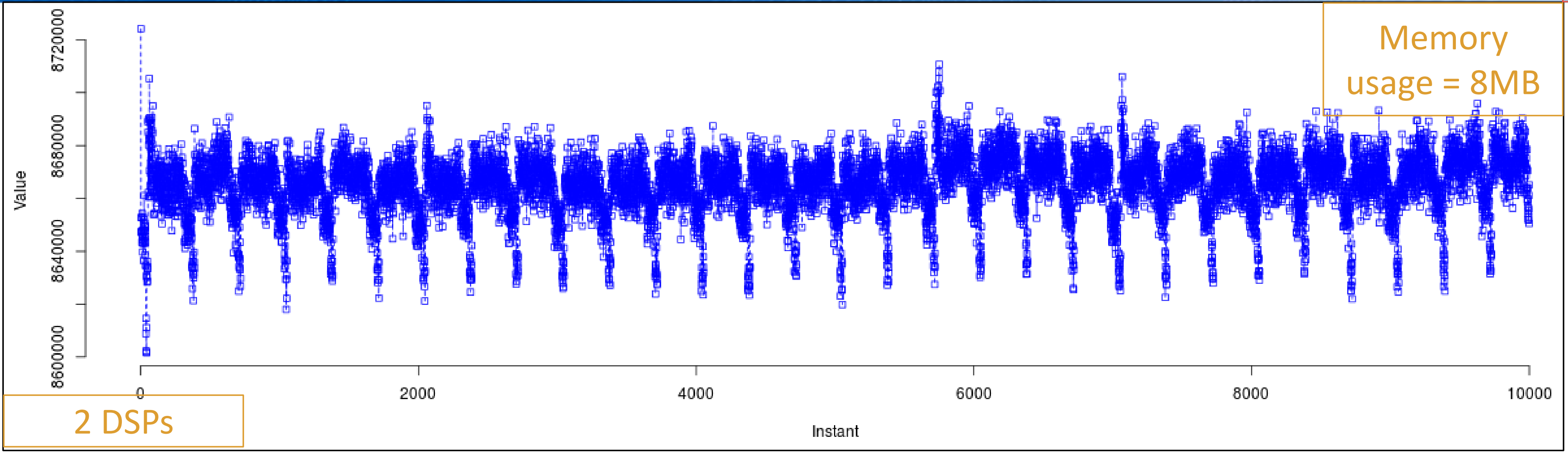


1 DSPs

Memory usage = 8MB

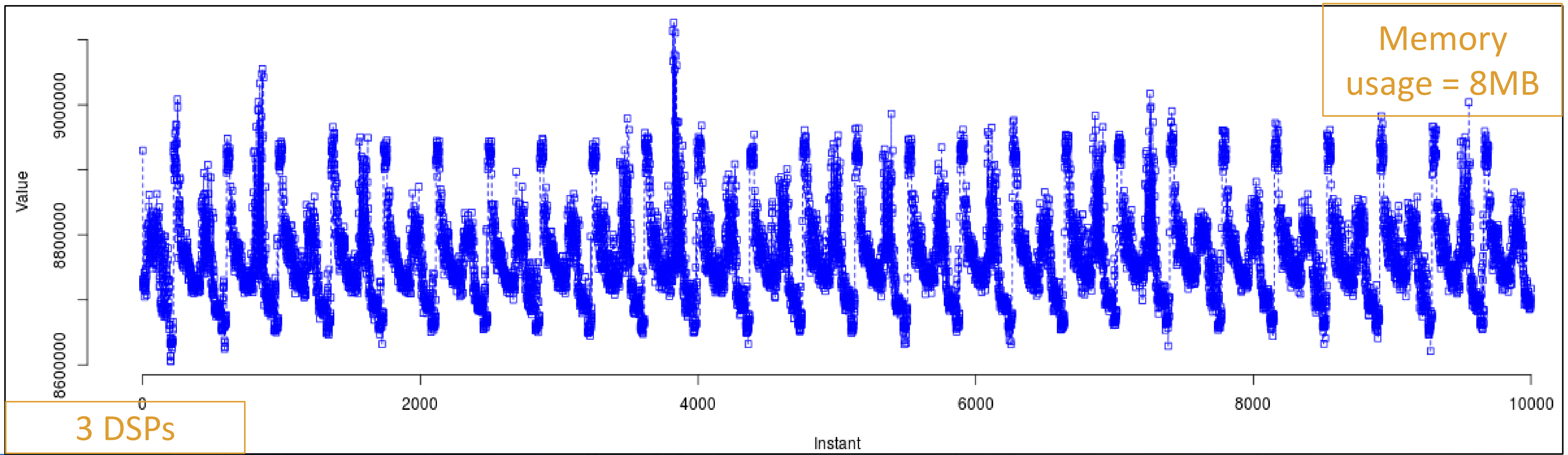
SCENARIO 3 RESULTS: EXECUTION CYCLES (SAFETY1)

Non-critical task memory usage = 12MB



Memory usage = 8MB

2 DSPs

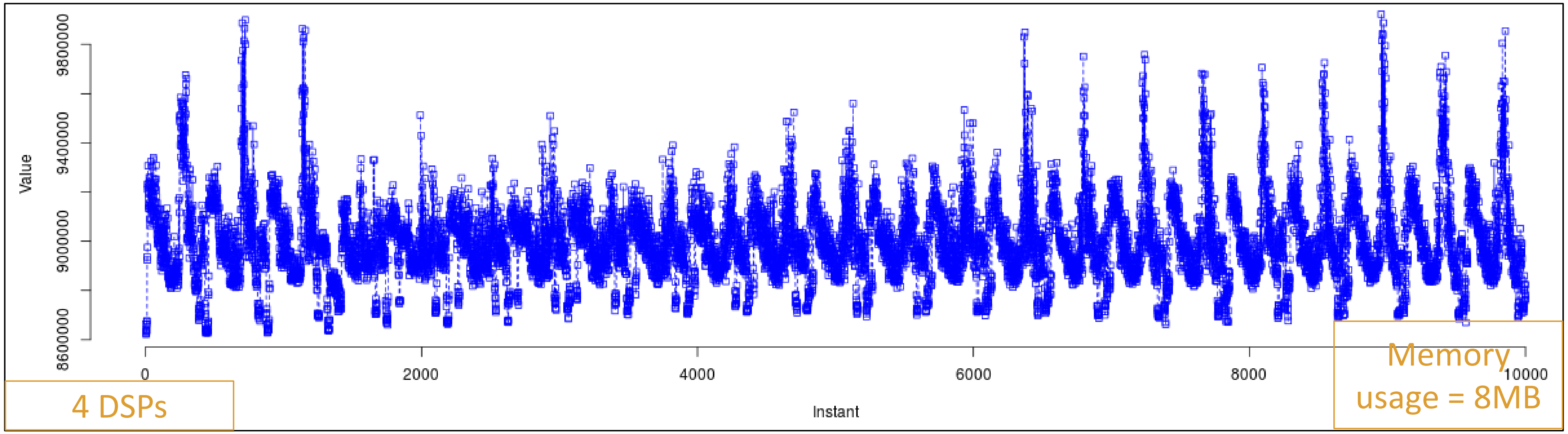


Memory usage = 8MB

3 DSPs

SCENARIO 3 RESULTS: EXECUTION CYCLES (SAFETY1)

Non-critical task memory usage = 12MB

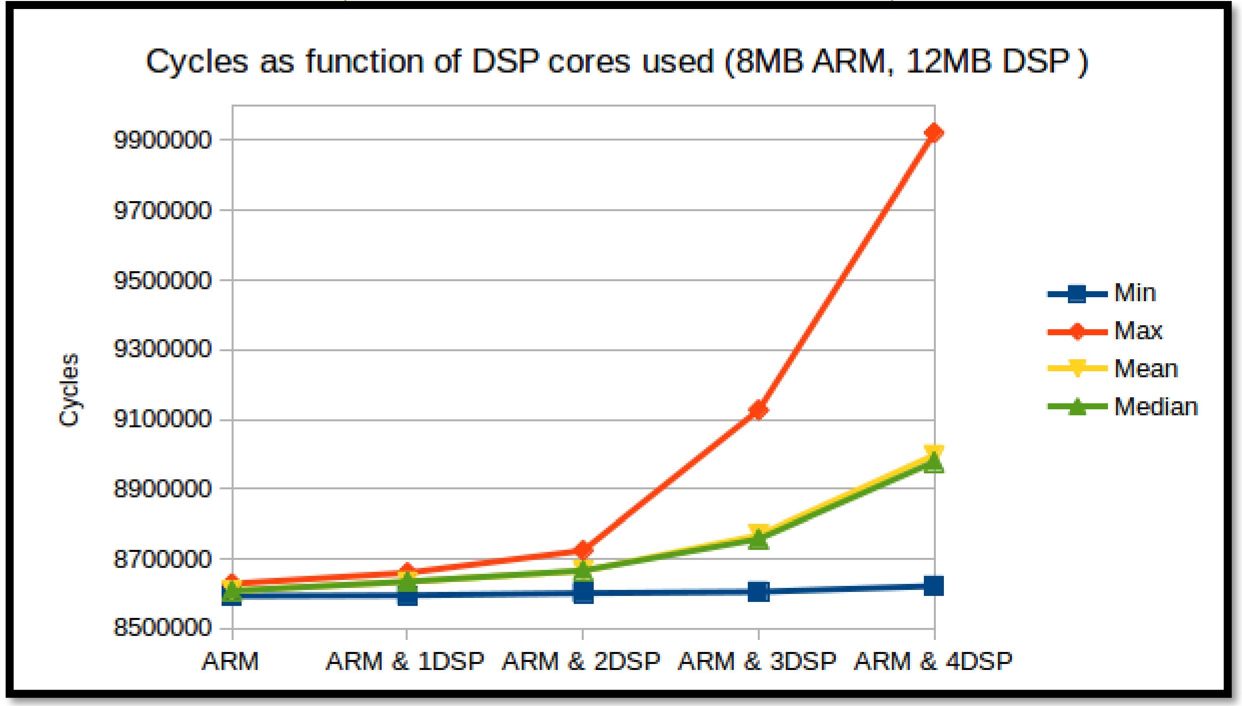


4 DSPs

Memory usage = 8MB

SCENARIO 3 SUMMARY: EXECUTION CYCLES

Safety1



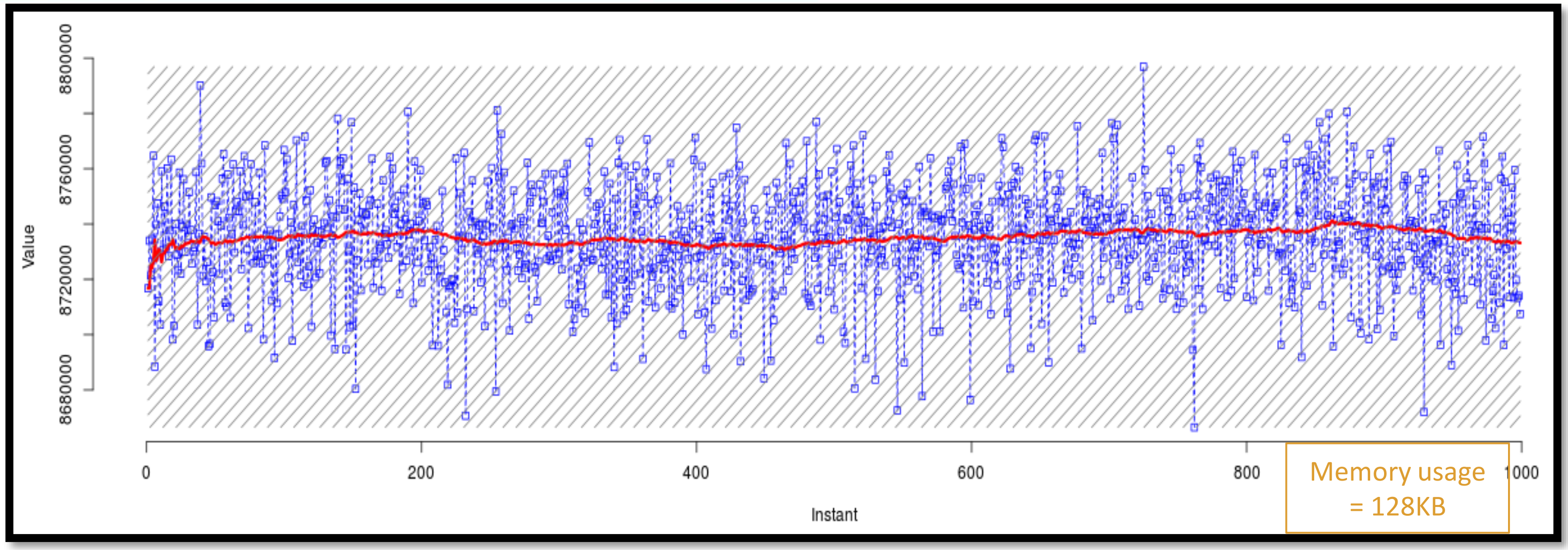
Cores	Mean Overhead (%)	Max Overhead (%)
ARM	0	0
ARM + 1DSP	0,299	0,363
ARM + 2DSP	0,659	1,105
ARM + 3DSP	1,854	5,769
ARM + 4DSP	4,514	14,991

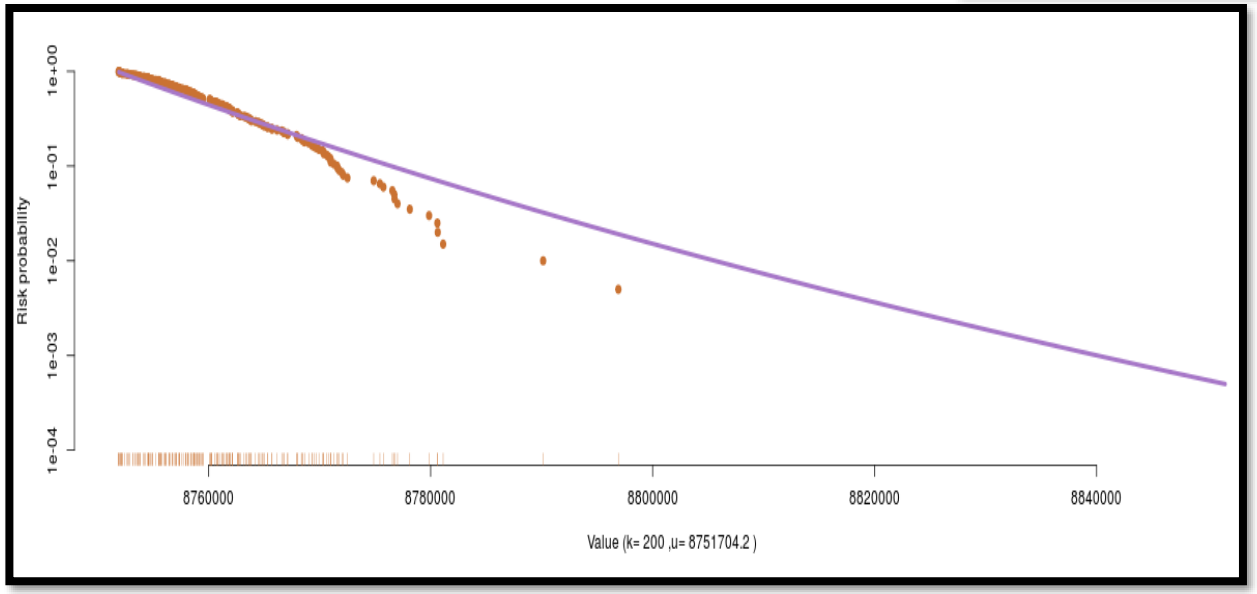
Data caches have been turned off

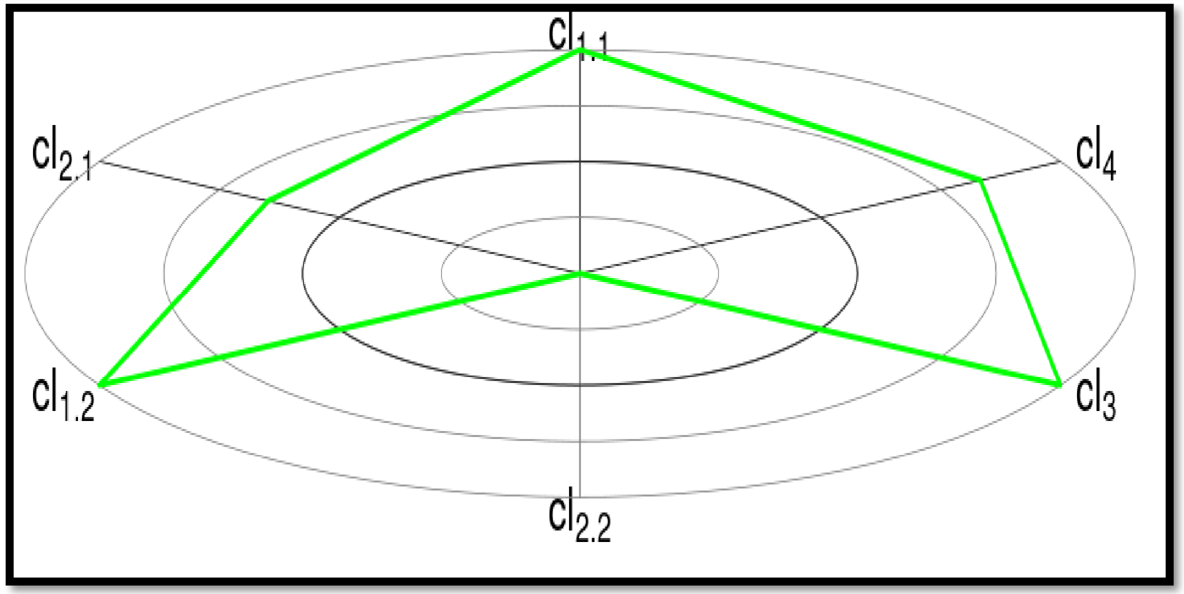
PREDICTABILITY

EVT application to the different scenarios

- Hypothesis check
- Inverse Cumulative Distribution Function (ICDF)
- Pay attention to its convergence







CONCLUSIONS

- Measurements based on Performance Monitor Hardware successfully works
- The EVT can successfully predict the outcome
- The best placement strategy is:
    1. The critical task in one ARM core
    2. Non-critical tasks in the DSPs (Resource accessing arbitration may be used if needed)
    3. Non-critical tasks in the second ARM (main interference source)