

Automatic Instrumentation of Dataflow Applications using PAPI

D. Madroñal¹, A. Morvan², R. Lazcano¹, R. Salvador¹, K. Desnos², E. Juárez¹, C. Sanz¹

¹Research Center on Software Technologies and Multimedia Systems, Universidad Politecnica de Madrid, Spain

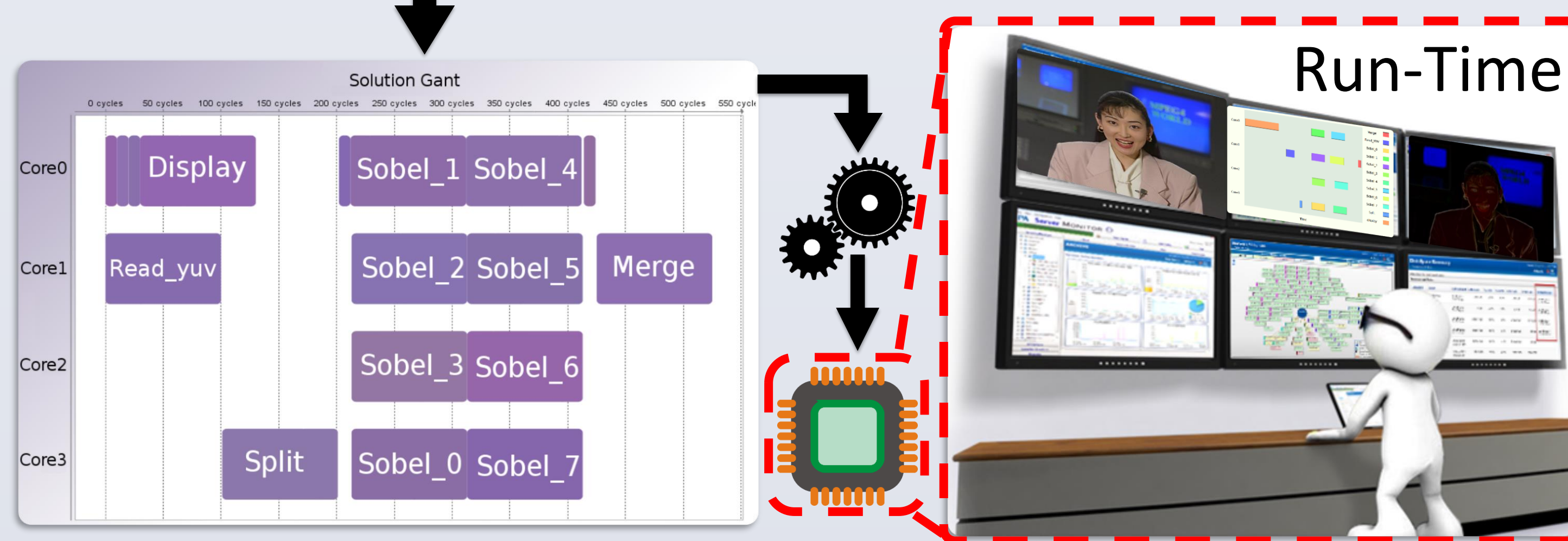
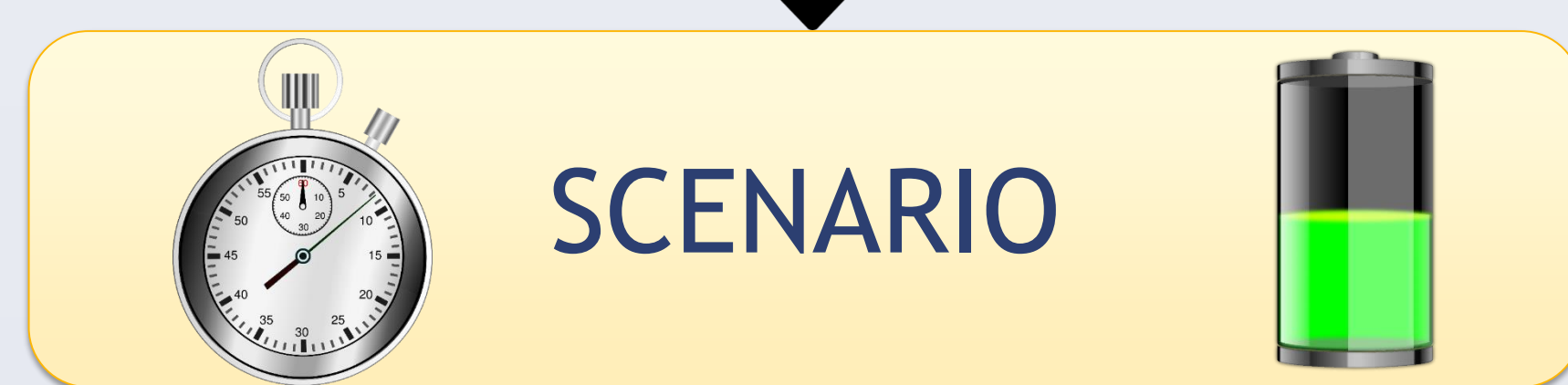
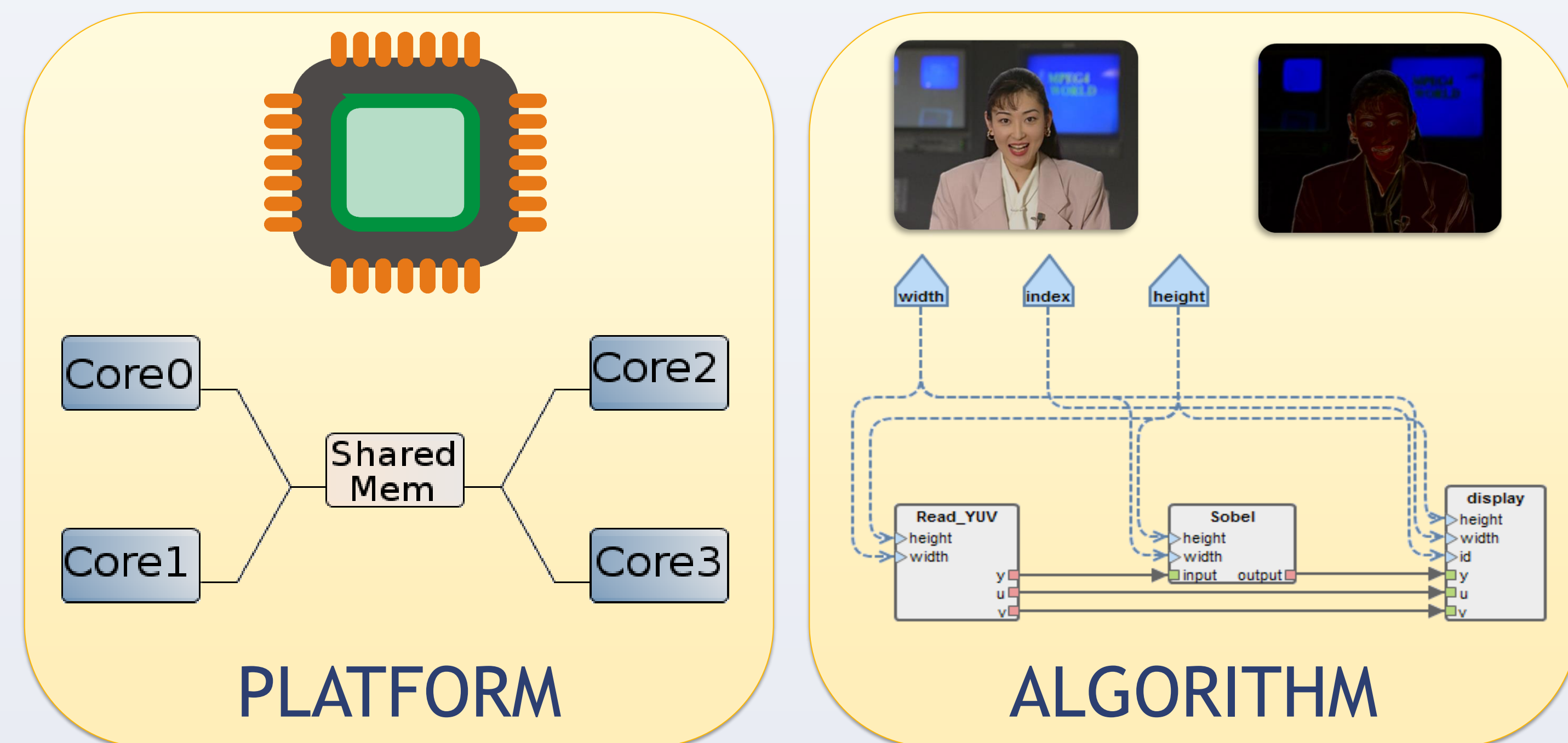
²Univ Rennes, INSA Rennes, CNRS, IETR – UMR 6164, Rennes, France

CF'18: Computing Frontiers 2018, Ischia, Italy

daniel.madronal@upm.es

INTRODUCTION

[] PREESM^[1]



TOOL

LIBRARY

PMC ACCESS

Papify

eventLib

PAPI

METHODOLOGY

a) Papify-PREESM user interface

Papify

The events needs to be associated to each core independently

Core0

PAPI components	Component type
☒ perf_event	CPU

Event Name	Short Description
☒ Timing	Event to time through PAPI_get_time()
☒ PAPI_L1_DCM	Level 1 data cache misses
☒ PAPI_L1_ICM	Level 1 instruction cache misses
☐ PAPI_L2_DCM	Level 2 data cache misses
☐ PAPI_L2_ICM	Level 2 instruction cache misses

b) Papify auto-generated monitoring

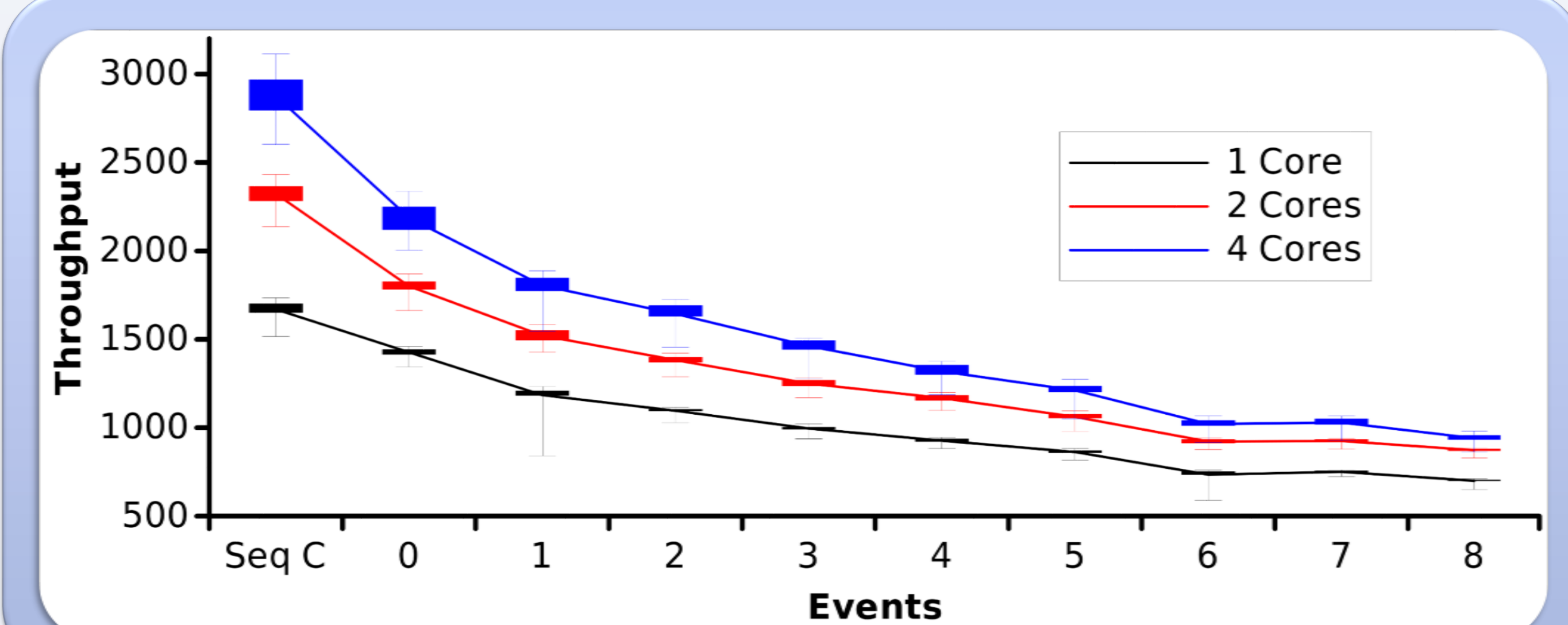
```
configure_papify(data_Read_YUV,PAPI_comp,PE_id,EventList);
event_start(data_Read_YUV,PE_id);
event_start_papify_timing(data_Read_YUV);
Read_YUV();
event_stop_papify_timing(data_Read_YUV);
event_stop(data_Read_YUV,PE_id);
event_write_file(data_Read_YUV);
```

1

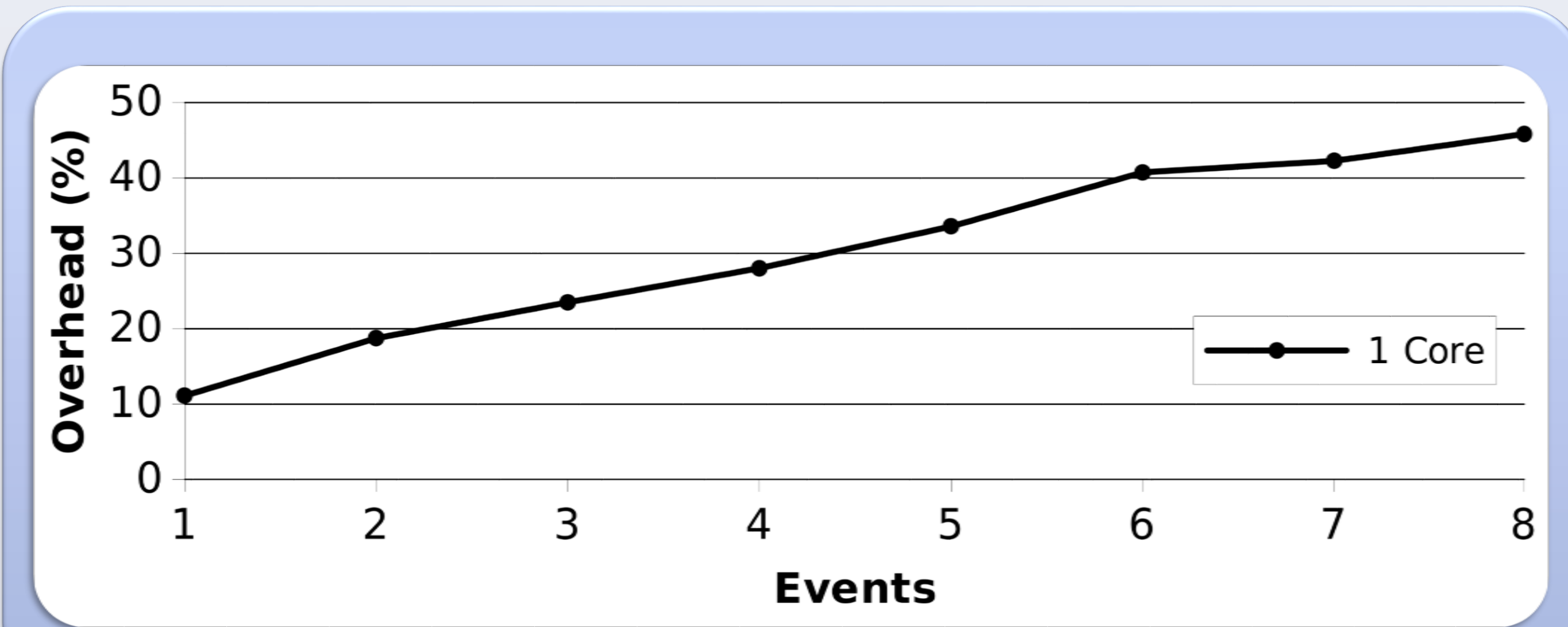
2

RESULTS

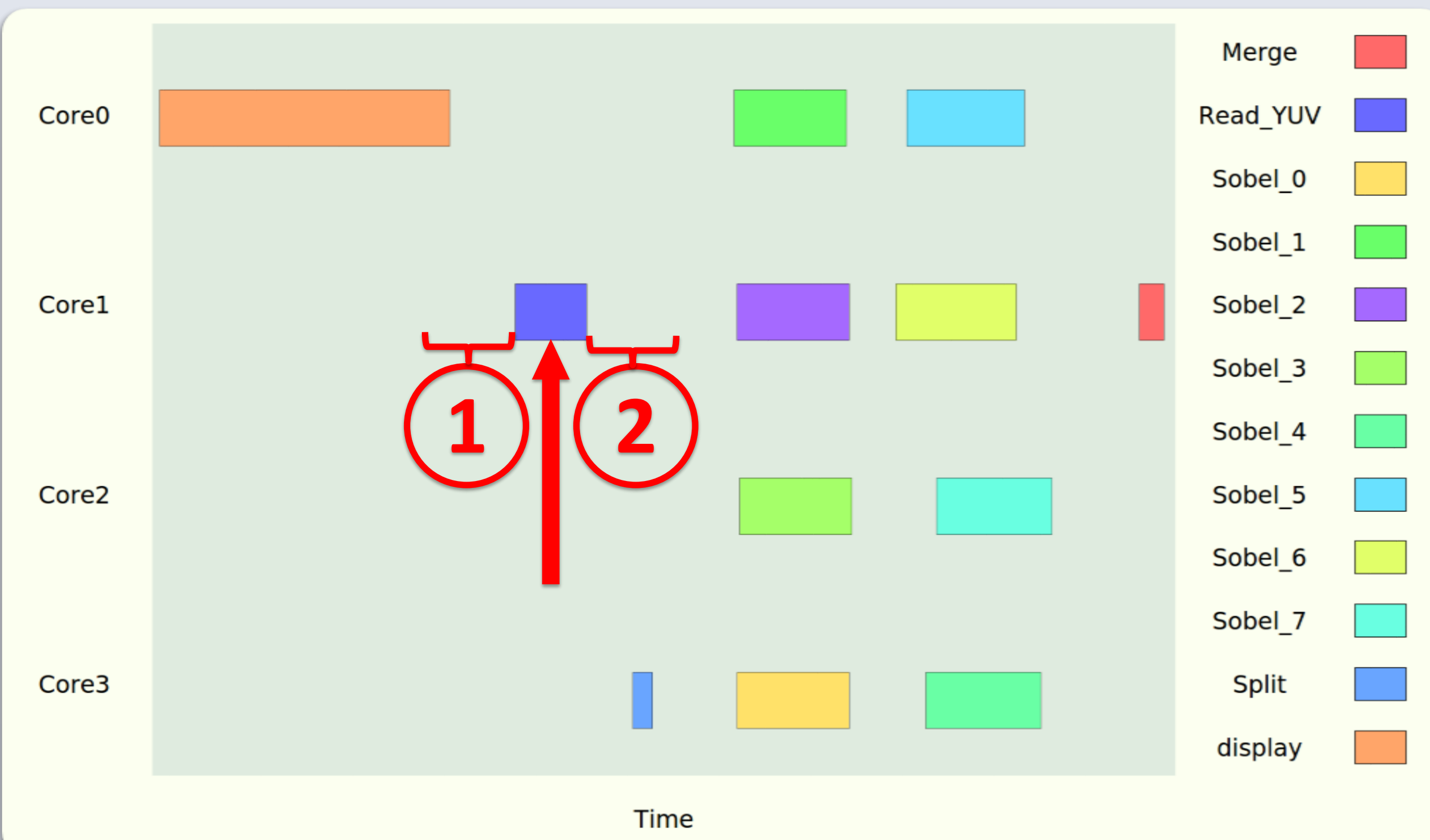
a) Papify impact on throughput - Sobel application



b) Papify execution time overhead - Sobel application



c) Papify-Viewer representation



FUTURE RESEARCH LINES

- To optimize the monitoring strategy
- To support monitoring by actor
- To include PAPI components for HW/SW targets
- To develop power/energy estimation models
- To include Papify in a real-time resource manager

ACKNOWLEDGMENTS

- ❖ Work funded by CERBERO project (H2020-732105)
- ❖ Papify-Viewer by Jaime Sancho (CITSEM - UPM)

[1] <http://preesm.insa-rennes.fr/website/>