Automatic Instrumentation of Dataflow Applications using PAPI

INTRODUCTION

METHODOLOGY

RESULTS

a) Papify impact on throughput - Sobel application

b) Papify execution time overhead - Sobel application

c) Papify-Viewer representation

FUTURE RESEARCH LINES

ACKNOWLEDGMENTS

The events needs to be associated to each core independently

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timing</td>
<td>Event to time through PAPI_get_time() Level 1 data cache misses Level 1 instruction cache misses Level 2 data cache misses Level 2 instruction cache misses</td>
</tr>
<tr>
<td>perf_event</td>
<td>Event time through PAPI_get_time()</td>
</tr>
</tbody>
</table>

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

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