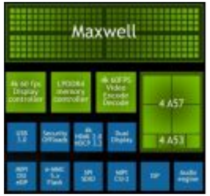
Experiences on the characterization of parallel applications in embedded systems with Extrae/Paraver

Adrian Munera, Sara Royuela,
Germán Llort, Estanislao Mercadal,
Franck Wartel, Eduardo quiñones

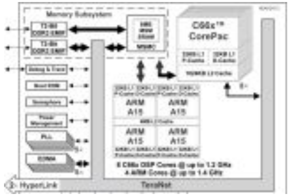
49th International Conference on Parallel Processing (ICPP2020)
17-20 August 2020, Edmonton, AB, Canada

Use of parallelism in embedded systems

- Demand for **high level of performance** in **embedded systems**.
- **Heterogeneity** introduces complexity to exploit **performance portability**.
- **Parallel programming models** are fundamental for **productivity**.



NVIDIA Tegra X1: 4-core A57/A53, GPU (automotive)



TI Keystone II: 4-core A15, 8-core DSP (industrial)

Kalray MPPA: four 4-core K1, 256-core fabric (avionics)

- **OpenMP** is an appropriate solution to leverage the potential of the architecture:
  - Provides **time-predictability**[1]
  - Shows delimited **correctness** guarantees [2]

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

[1] Serrano et. al, *Timing characterization of OpenMP4 tasking model*. CASES 2015.

[2] Royuela et. al, A Functional Safety OpenMP* for Critical Real-Time Embedded Systems. IWOMP 2017.

Analyzing parallelism in embedded systems

- Parallelism affects functional and non-functional behavior (time, energy, memory, etc.)
- Need to analyze the **impact of parallelism** on the functional (**FR**) and non-functional (**NFR**) requirements.

Analysis tool domain	Parallel programming model	Performance	NFR
HPC	✓	✓	✗
Embedded	✗	✓	✓

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Analysis tools: classification

Data gathering method

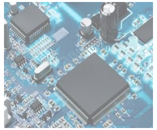
	✓	✗
Basic measurements	Easy to obtain	Come without information about factors
Sampling	Provide better understanding of the application	Cannot characterize fine-grained tasks
Instrumentation	Captures the activity as it is	May introduce overhead

Data storage method

	✓	✗
Profiling	Produce a summary of the picture	Lack information for specific points in time
Tracing	Capture exact picture	May introduce overhead

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Analysis tools: from embedded to HPC systems



EC

Hardware solution

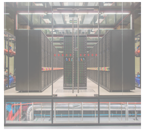
- ❖ **ULINKplus** Debug Adapter
  - ➢ µVision IDE
- ❖ **J-Trace** Debug Probe
  - ➢ SystemView analyzer

Timing behavior

- ❖ **RapiTask**
- ❖ **RapiTime**

OS behavior

- ❖ **LTTng**
- ❖ **Tracealyzer**



HPC

Compile-time instrumentation

- ❖ **Score-P**
  - ➢ Scalasca
  - ➢ Vampire
  - ➢ TAU

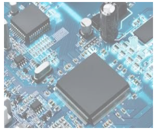
Compile- and run-time instrumentation

- ❖ **Extrae**[1]
  - ➢ Paraver

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

[1] https://tools.bsc.es/extrae

Analysis tools: from EC to HPC systems



EC

Hardware solution

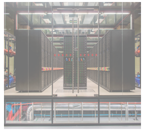
- **ULINKplus** Debug Adapter
  - µVision IDE
- **J-Trace** Debug Probe
  - SystemView analyzer

Timing behavior

- **RapiTask**
- **RapiTime**

OS behavior

- **LTTng**
- **Tracealyzer**



HPC

Compile-time instrumentation

- **Score-P**
  - Scalasca
  - Vampire
  - TAU

Compile- and run-time instrumentation

- **Extrae**[1]
  - Paraver

✓ **Sampling** ✓ **Tracing**
✓ **Instrumentation** ✓ **Profiling**

✓ **Parallel model characterization**
✗ **Non-functional requirements**

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

[1] https://tools.bsc.es/extrae

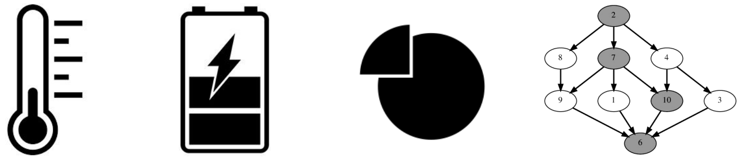
Proposal: adapting Extrae to EC systems

Adapt to a embedded system

1. Static environment
2. RTOS
3. Specific architecture modules

Analyze NFR

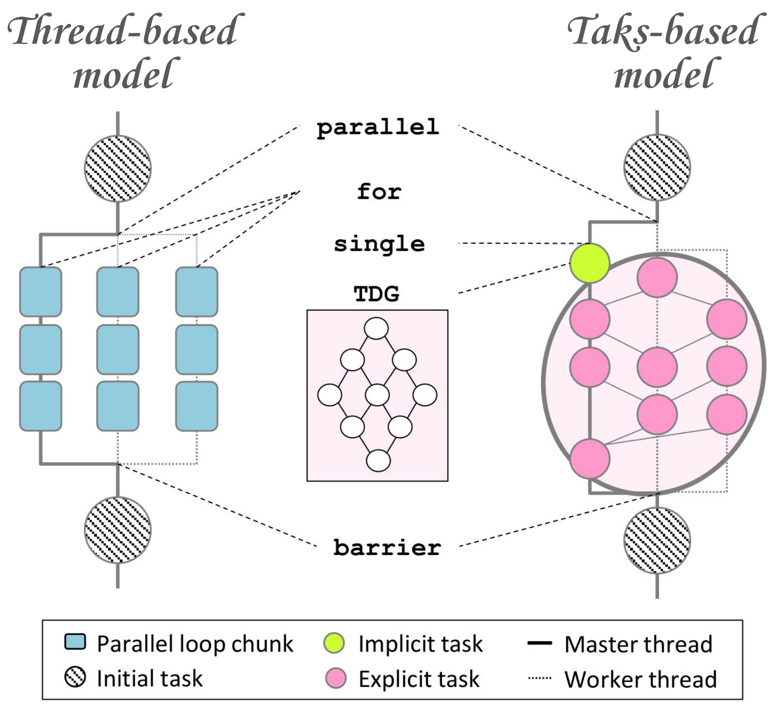
1. Temperature and power consumption
2. Memory consumption
3. Tasks communication



Outline

- The characterization of OpenMP
- Accommodating Extrae to embedded systems: the GR740
- New functionalities in Extrae
- Analysis: correlating parallelism and non-functional requirements
- Conclusions

The characterization of OpenMP



Parallel Programming Model

- ➔ Exposed parallelism
- ➔ Load balance
- ➔ Synchronization overhead
- ➔ Contention overhead

Non-functional requirements

- ➔ Performance
- ➔ Power consumption
- ➔ Temperature

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Embedded Systems: the GR740

Radiation-hard SoC designed as the ESA Next Generation Microprocessor.

Hardware

- LEON4 SPARC V8 @250MHz
- IEEE-754 floating point unit
- 16KB instruction and data caches
- 2MB write-back L2 cache
- LEON4 Statistics Unit, L4stat
- AHB Bus
- Temperature sensor controller
- Timer units

Software

- RTEMS RTOS
- RCC cross compilation system
- RTEMS-5.0 C/C++ real-time kernel with support for SMP
- Newlib
- L4stat driver

Adapting Extrae to the GR740

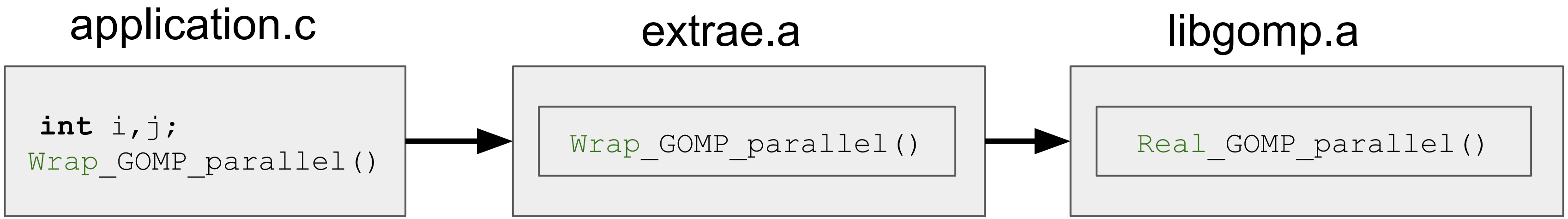
1. Intercepting calls in a static environment
2. POSIX dependence
3. Retrieving function names
4. Trace generation
5. Supporting hardware counters
6. Statically defining the environment

Adapting Extrae to the GR740

1. Intercepting calls in a static environment:

   OpenMP Call → Extrae → OpenMP runtime

   - **Vanilla Extrae:** LD_PRELOAD mechanism at runtime.
   - **Adapted Extrae:** Symbol wrapping at compile time, using linker flags.

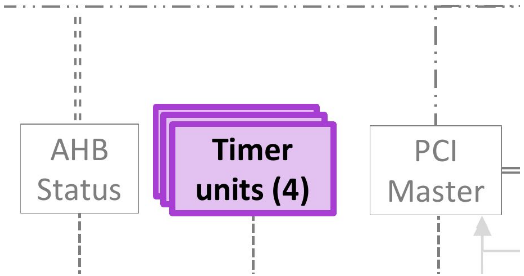


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Adapting Extrae to the GR740

2. **POSIX** dependence:

- Extrae relies on standard functions and structures from **POSIX**.
- Unfortunately, not all C standard libraries implement all **POSIX** functions.
- **Newlib** does not implement the **ucontext** structure, used for implementing the sampling mechanism. In the adaptation it has been replaced by hardware timers.

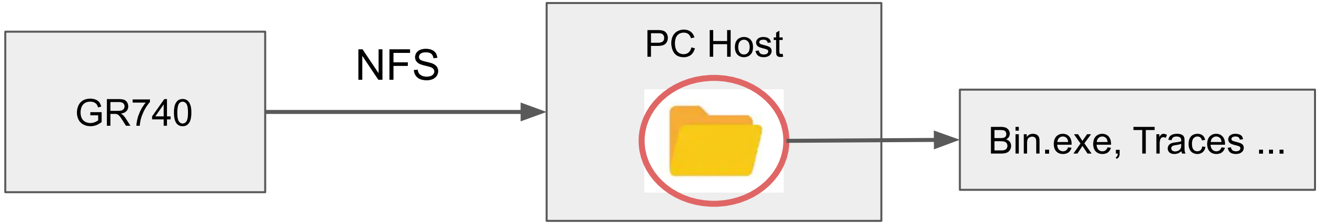


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Adapting Extrae to the GR740

3/4. Retrieving function names and trace generation:

- Originally, **Extrae** obtains the symbol names of the executable using the **binutils libraries** targeting the binary from the file system.
- The binary is not available inside the board file system, since it is loaded in RAM. In the adaptation, **Extrae** now specifies the binary path and the use of a remote file system (**NFS**).
- This remote file system is also required for **generating the final traces**, where we also need to take into account the file system limitations (maximum file size, maximum size per write, etc)

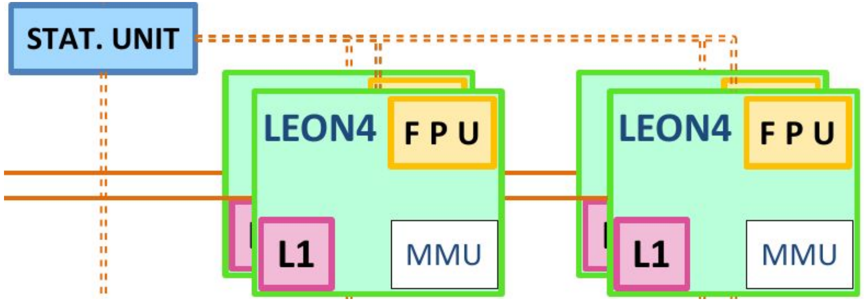


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Adapting Extrae to the GR740

5. Supporting hardware counters:

- Vanilla Extrae relies on **PAPI** library to gather the hardware counters of the system. **PAPI** does not support the **GR740** architecture.
- The **GR740** board provides the **L4STAT** unit, that implements hardware counters. This data is accessible through the **L4STAT** driver.
- We have extended **Extrae** to additionally support the **L4STAT** driver instead of just **PAPI**.



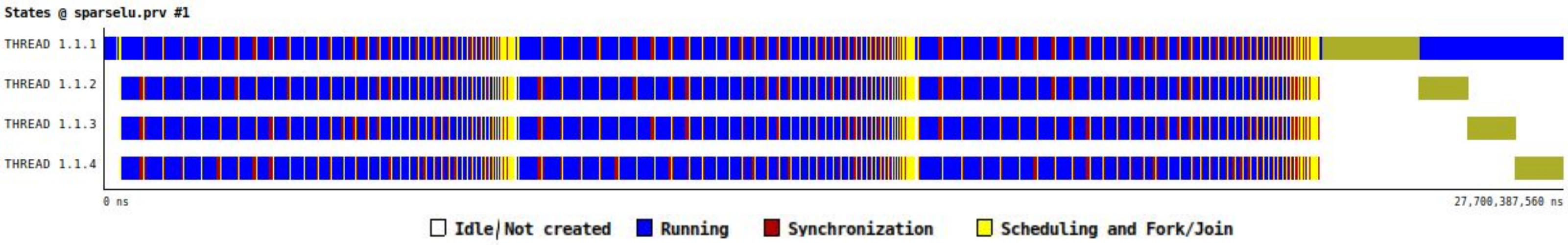
Analysis: Applications & Aspects

Applications	Evaluated aspects
SparseLU loops	Memory: stack and heap Temperature and power consumption
SparseLU tasks	Task communication
Image processing	Sampling

Analysis: SparseLU

SparseLU loops

```
#pragma omp parallel private(kk)
for (..) // 3 iterations
#pragma omp single
        lu0(BENCH[kk*bots_arg_size+kk]);
#pragma omp for nowait schedule(dynamic)
 for(..)
        fwd(BENCH[kk*bots_arg_size+kk], BENCH[kk*bots_arg_size+jj]);
#pragma omp for schedule(dynamic)
 for (...)
         bdiv (BENCH[kk*bots_arg_size+kk], BENCH[ii*bots_arg_size+kk]);
…….
```

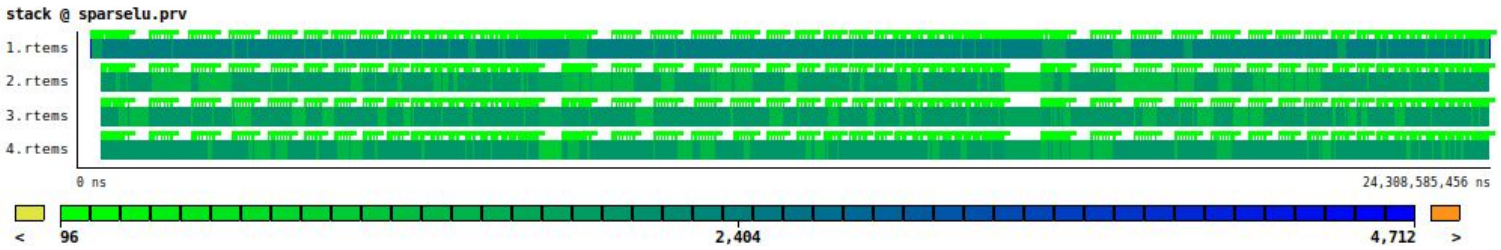


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Analysis: memory consumption

Runtime states

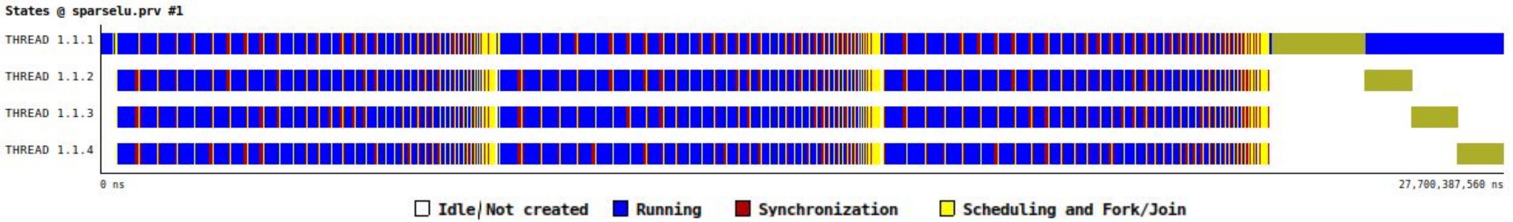
Stack



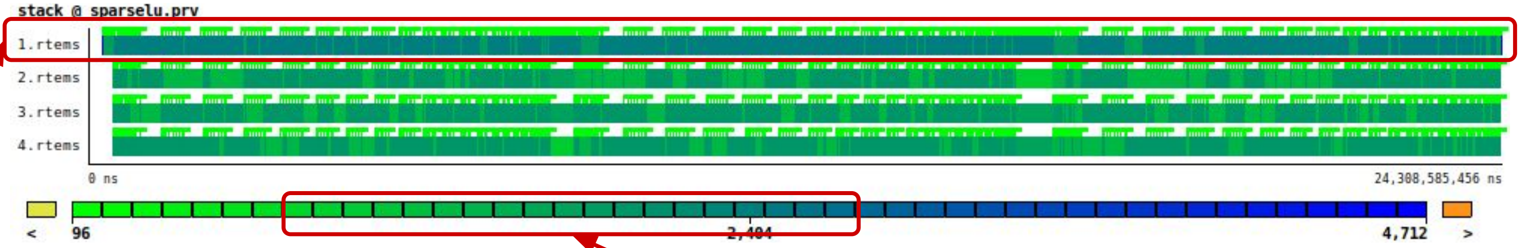
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Analysis: memory consumption

Runtime states



Stack



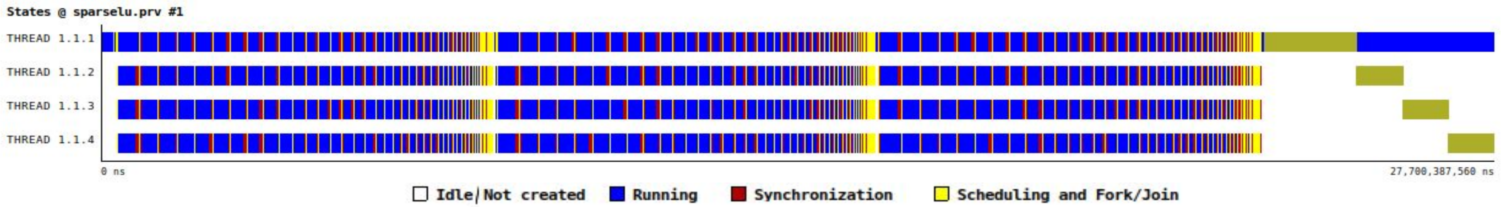
The main thread uses more stack memory than the others.

Application uses stack size between 1000 and 3000

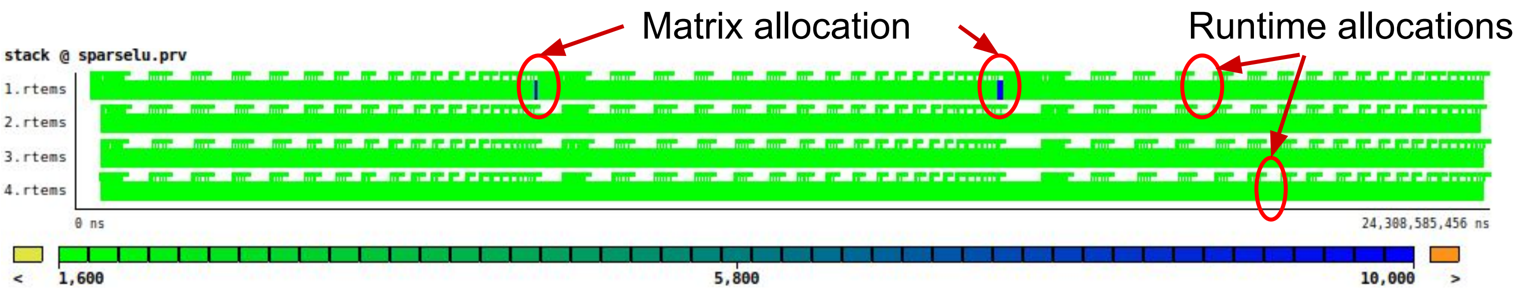
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Analysis: memory consumption

Runtime states

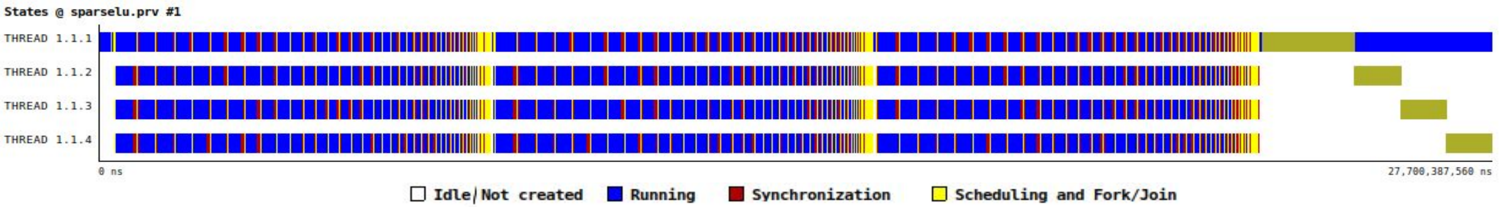


Dynamic (de) allocation

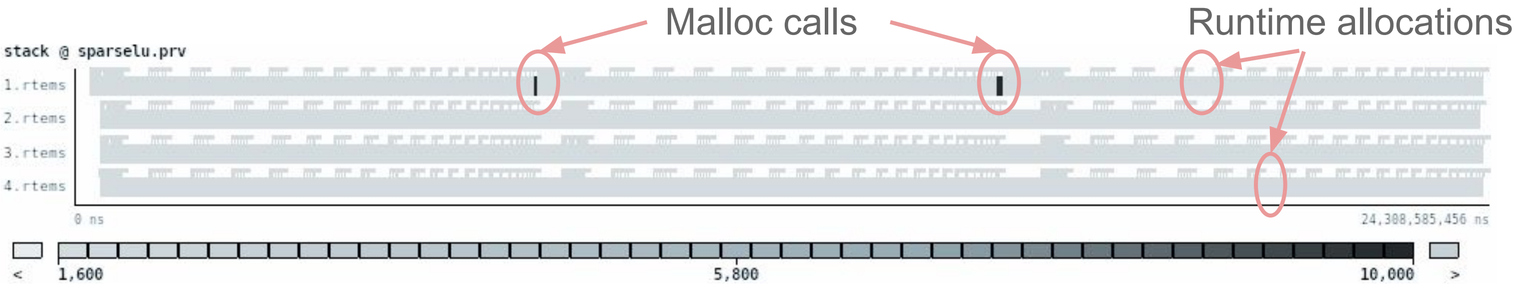


Analysis: memory consumption

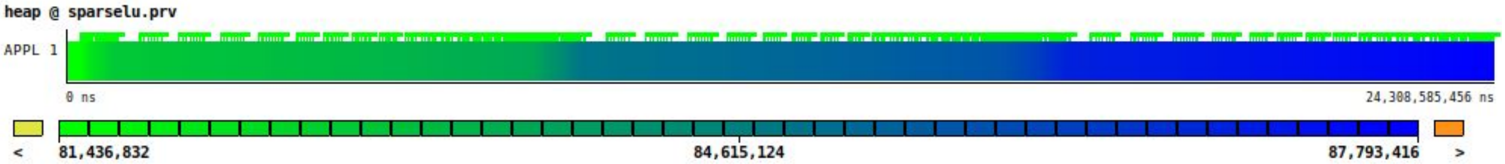
Runtime states



Dynamic (de) allocation



Heap



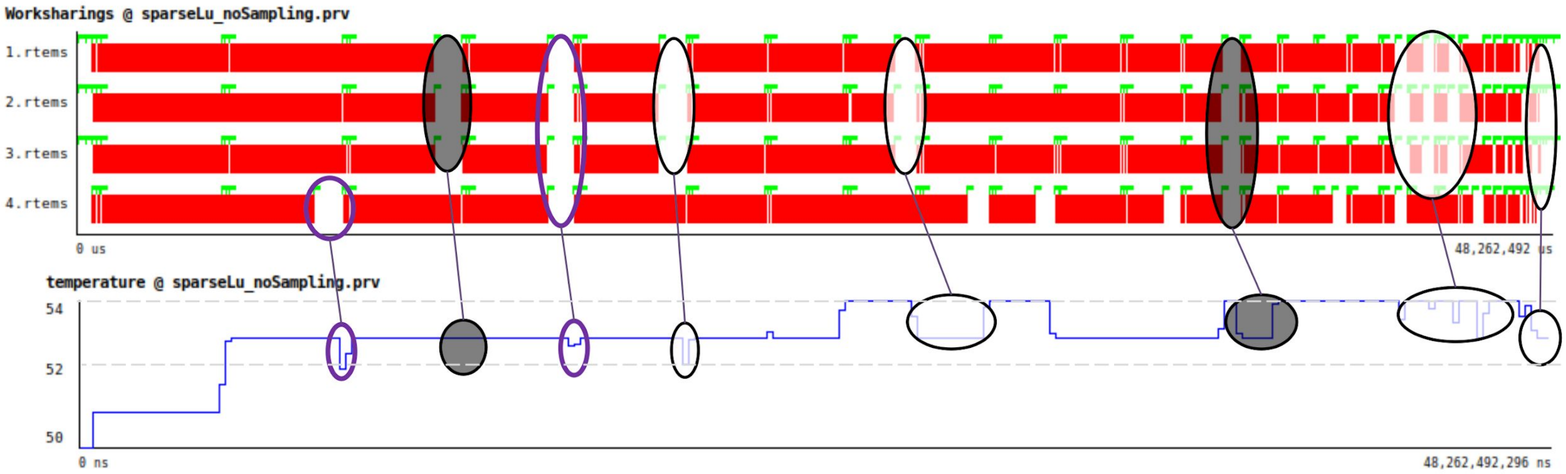
Heap does not decrement, since memory does not return to the OS although it is freed.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Analysis: temperature

Work sharings

Temperature



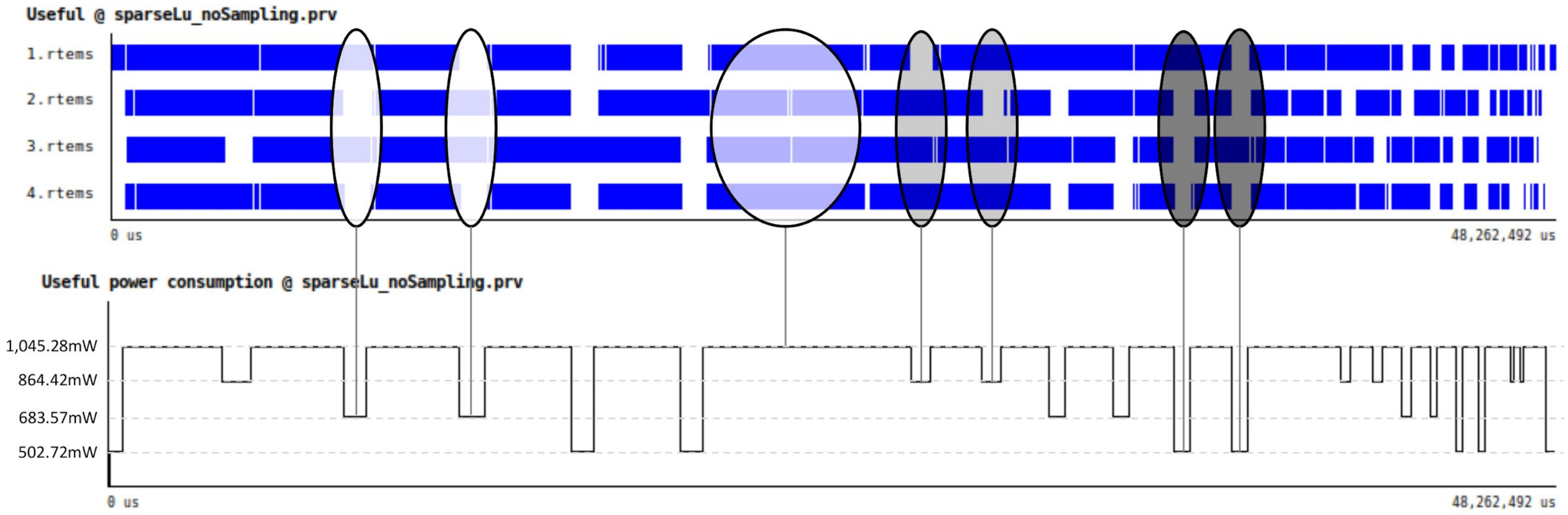
The temperature of the system is correlated with the cpu usage.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Analysis: power consumption

Parallel execution

Power consumption



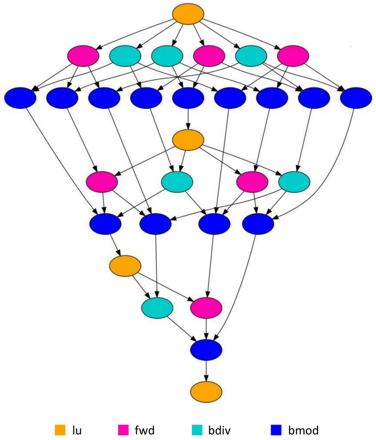
The power consumption can be calculated using the information about cpu usage.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

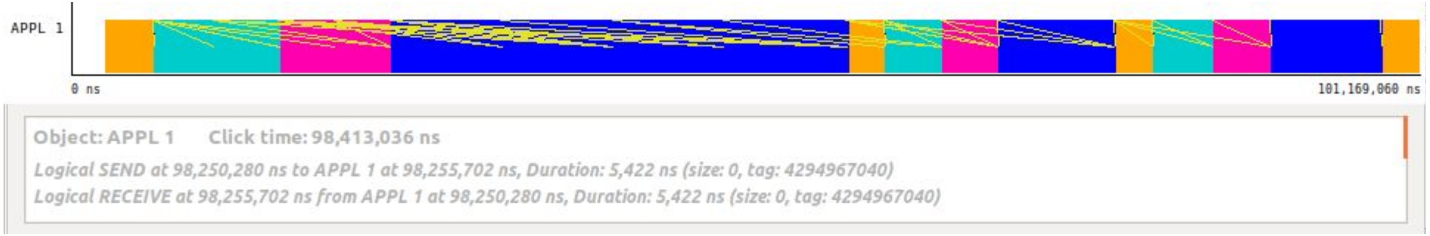
Analysis: tasks communication

SparseLU tasks

TDG



Task communication



Tasks dependencies can be represented inside the traces.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

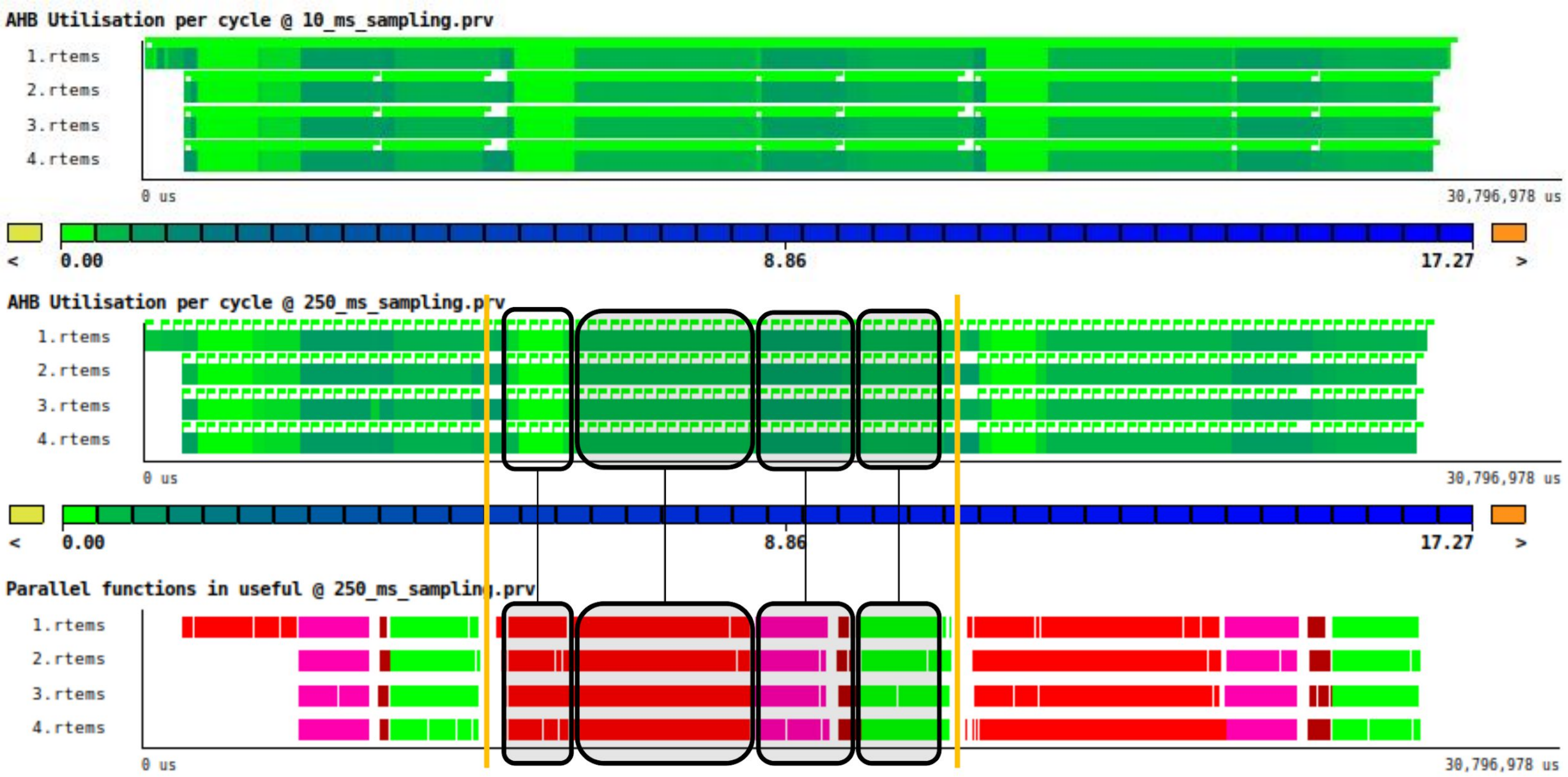
Analysis: sampling and the AMBA bus

Image processing

Sampling 10ms

Sampling 250ms

Parallel user functions



Extrae extensions portability

Extensions	Applicable to
1. Temperature and power consumption	GR740 boards
2. Memory consumption	RTEMS operating systems
3. Tasks communication	OpenMP-compatible systems

Conclusions

- Currently embedded systems lack of tools to analyze applications performance at parallel programming level.
- HPC analysis tools do not support the analysis of non-functional requirements.
- Well-tested performance tools such as Extrae can be:
  - adapted to the constraints of embedded systems, e.g., RTEMS + GR740.
  - extended to analyze non-functional requirements, such as temperature and power consumption, a key aspect in embedded systems.

Experiences on the characterization of parallel applications in embedded systems with Extrae/Paraver

adrian.munera@bsc.es

Work partially funded from the HP4S (High-Performance Parallel Payload Processing for Space) project under ESA-ESTEC ITI contract Nº 4000124124/18/NL/CRS

ICPP2020