Development environment for multicore processors
Task 3 Demonstrator implementation

Fabrice Cros
Final Presentation Days 01/06/2015

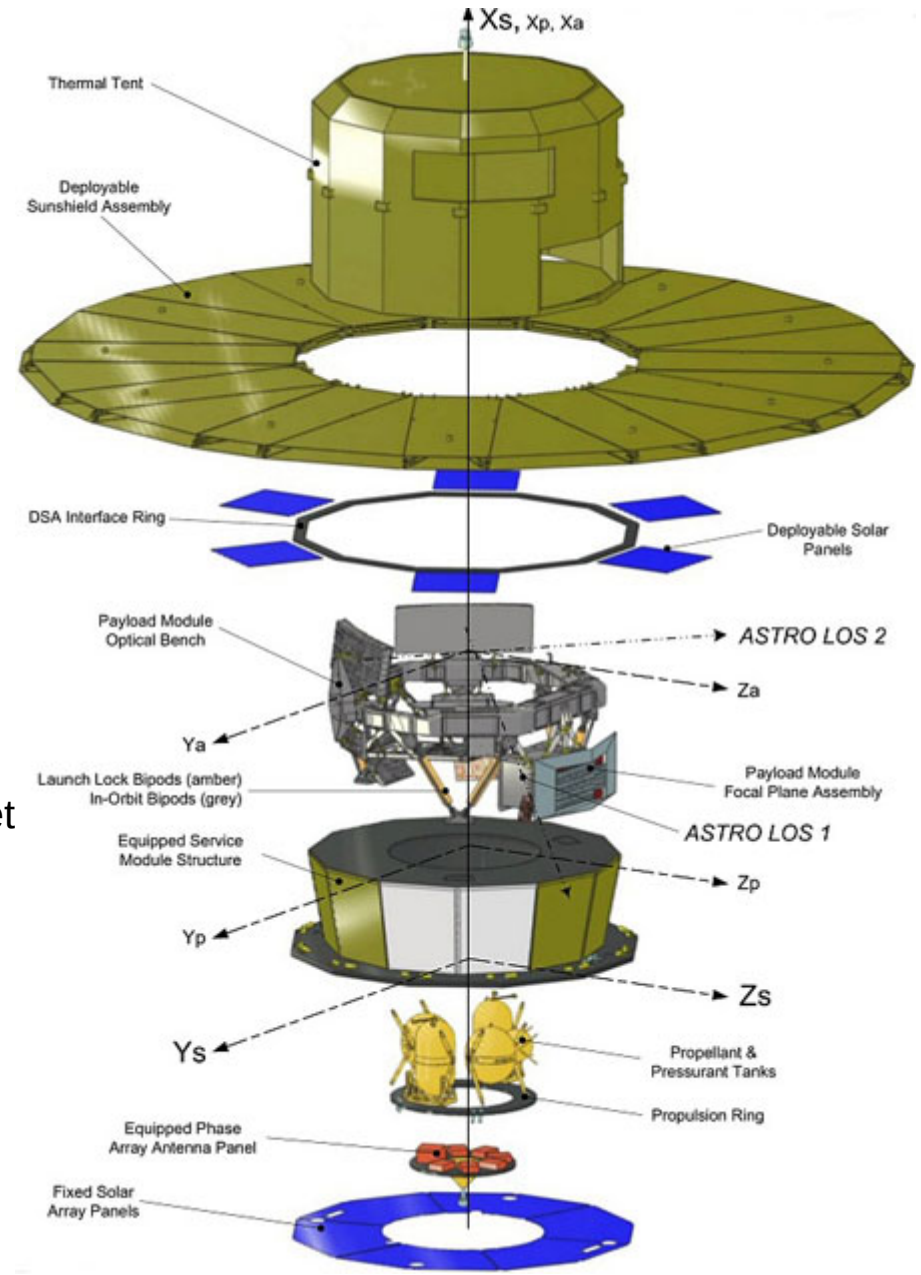
AIRBUS DEFENCE & SPACE

GAIA Satellite

2 optical telescopes
1 focal plane populated with CCDs
Complex image processing: 49 algorithms pipelined

GAIA VPU functionalities:

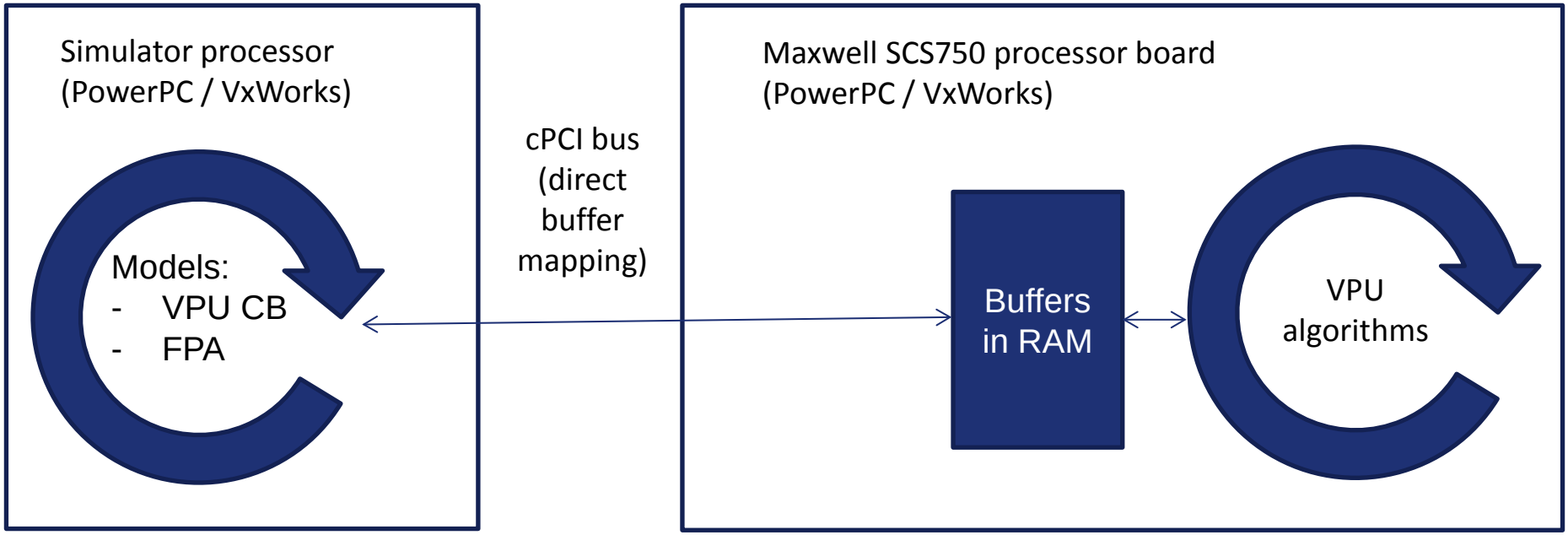
- Commanding of the CCDs and data collection
- Detection of potential astronomical objects
- Selection of objects to observe
- Confirmation of objects
- Collection of scientific data and formation of star data packet
- Transfer of data packets to the Payload Data Handling Unit
- Supply of star velocity information to the AOCS subsystem
- Collection of FPA and VPU housekeeping



AIRBUS DEFENCE & SPACE

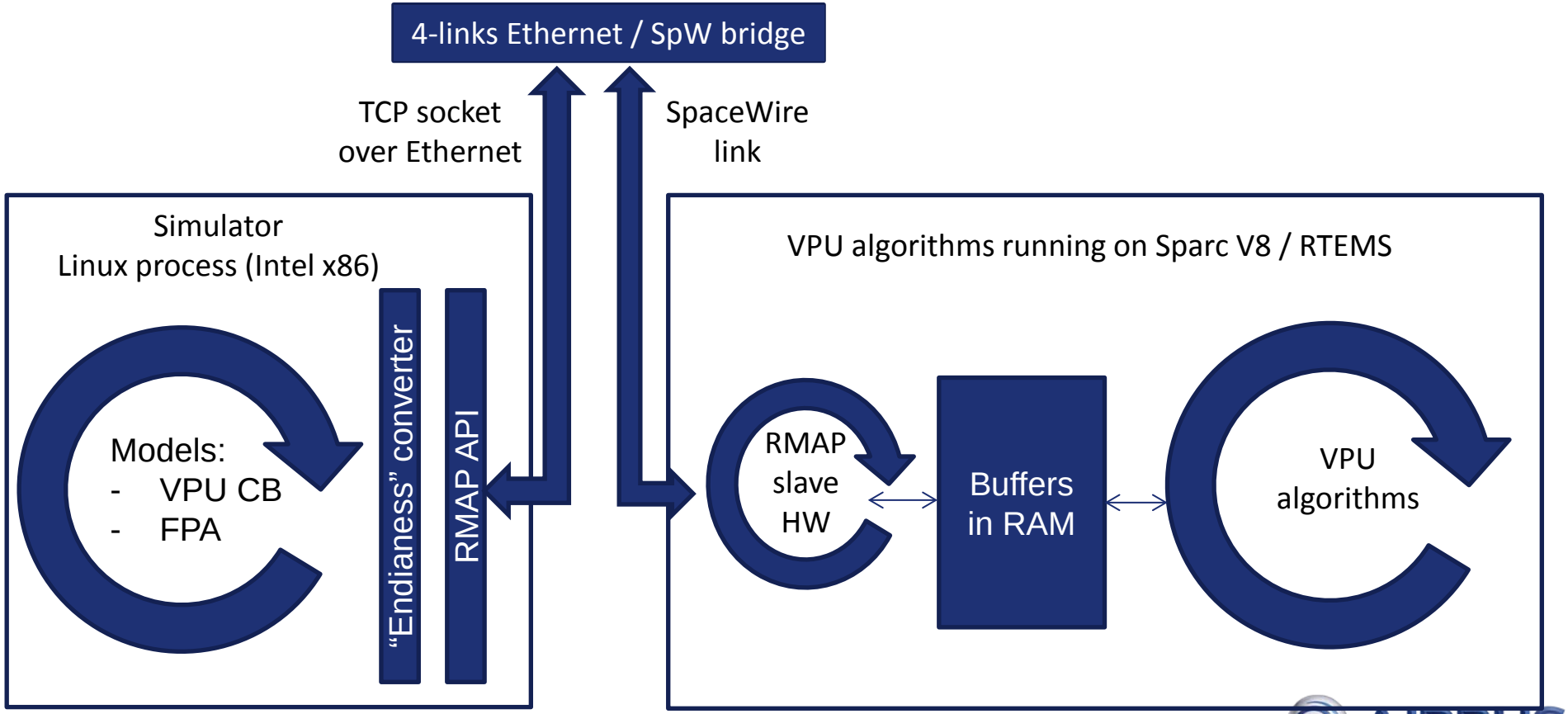
Demonstrator starting point

- GAIA VPU application running on PPC / VxWorks
- Simulator running on PPC VxWorks
- Communication via PCI bus , DMA accesses



Demonstrator setup for NGFP and GR712 boards

- Porting of GAIA VPU to run on SparcV8/RTEMS
- Data compression was removed from GAIA application (code not portable)
- Simulator running on linux (little endian)
- DMA “emulated” via SpW RMAP



Demonstrator platforms

NGFP

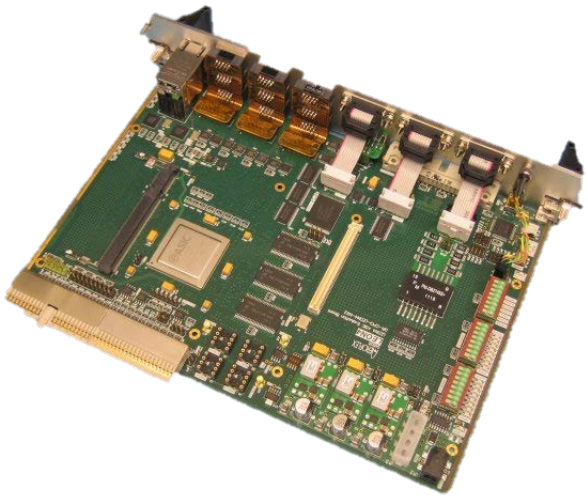
4 Cores

Configuration 1:

- CPU 150MHz
- DDR memory 300MHz

Configuration 2:

- CPU 200MHz
- DDR memory 300MHz



GR712

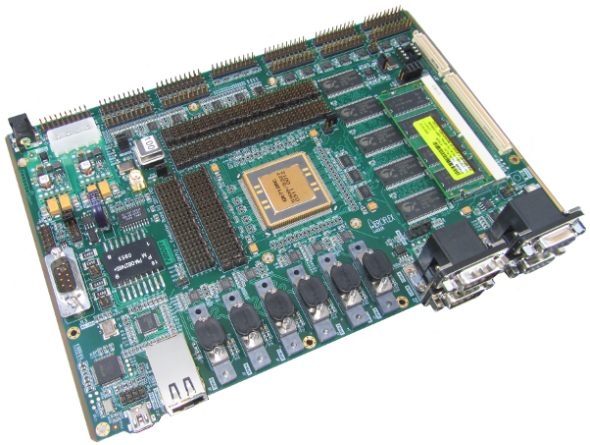
2 Cores

Configuration 1:

- CPU 48MHz
- SDRAM memory 48MHz

Configuration 2:

- CPU 80MHz
- SDRAM memory 80MHz



AIRBUS DEFENCE & SPACE

Parallelization scheme

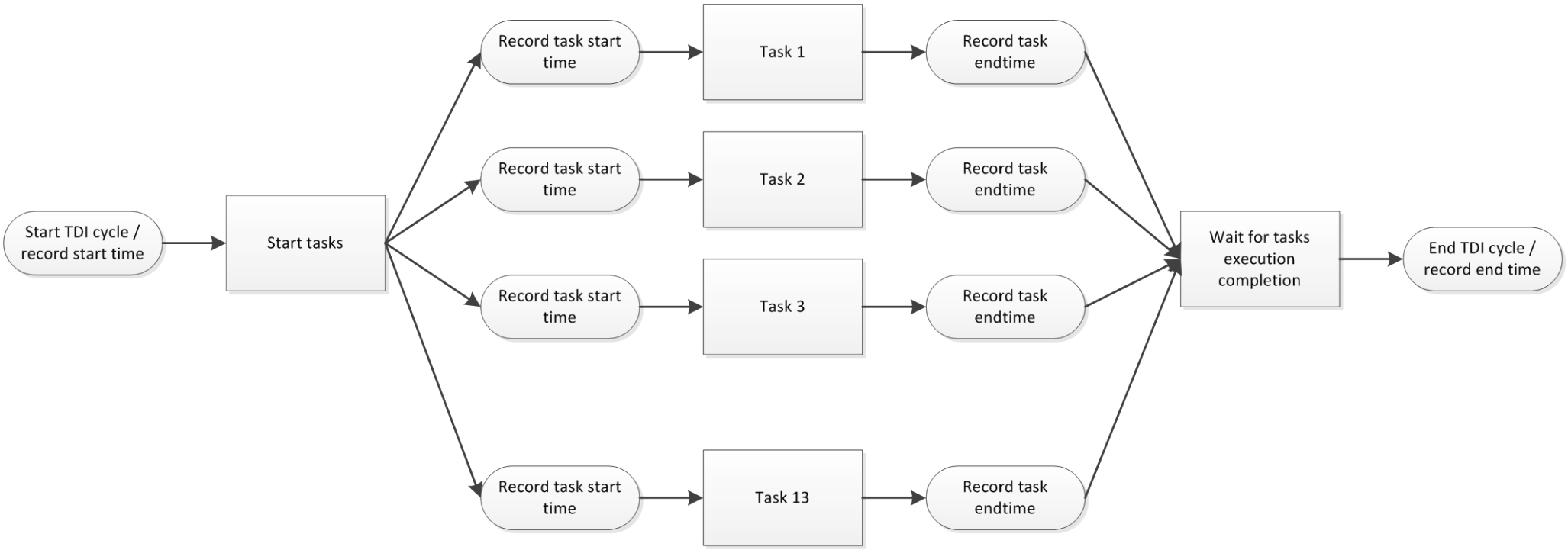
- **Task parallelism**
  - Grouping of 49 processing functions in 13 tasks that can be executed concurrently
  - Tasks are working on independent data sets
  - At each TDI cycle, all 13 tasks are run. We wait for completion of all tasks before next TDI cycle.
  - Shared resources are protected by RTEMS semaphores

- **MTAPI usage**
  - Each processing task is registered as a MTAPI action.
  - Each core is an MTAPI node
  - Main node spawns all tasks for one cycle and then start executing actions as well

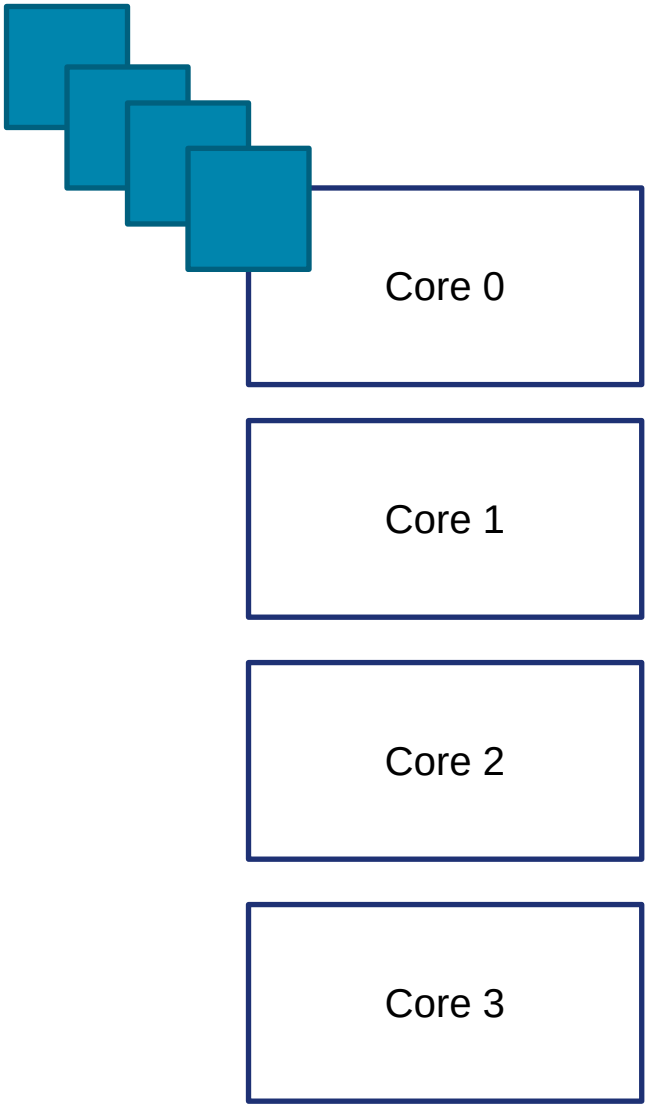
- **Timing measurements**
  - Duration of each task is recorded for each TDI cycle using high resolution timer

AIRBUS DEFENCE & SPACE

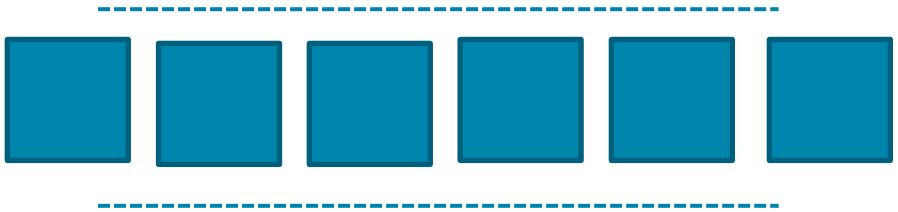
Task parallelism



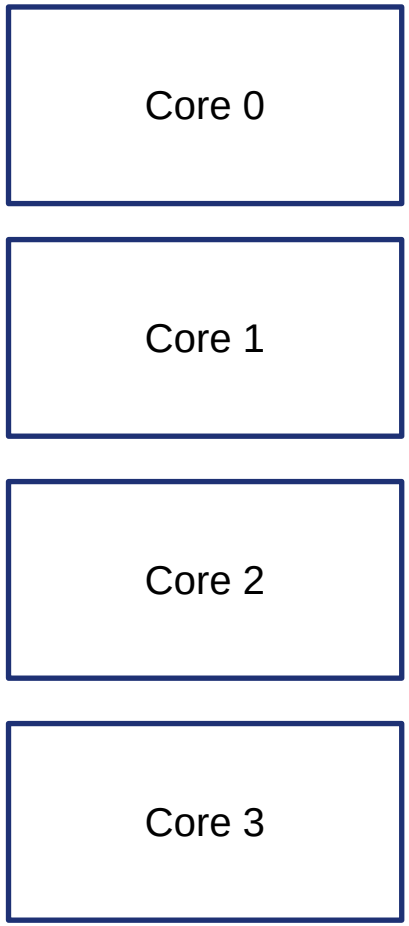
MTAPI Usage



MTAPI Usage

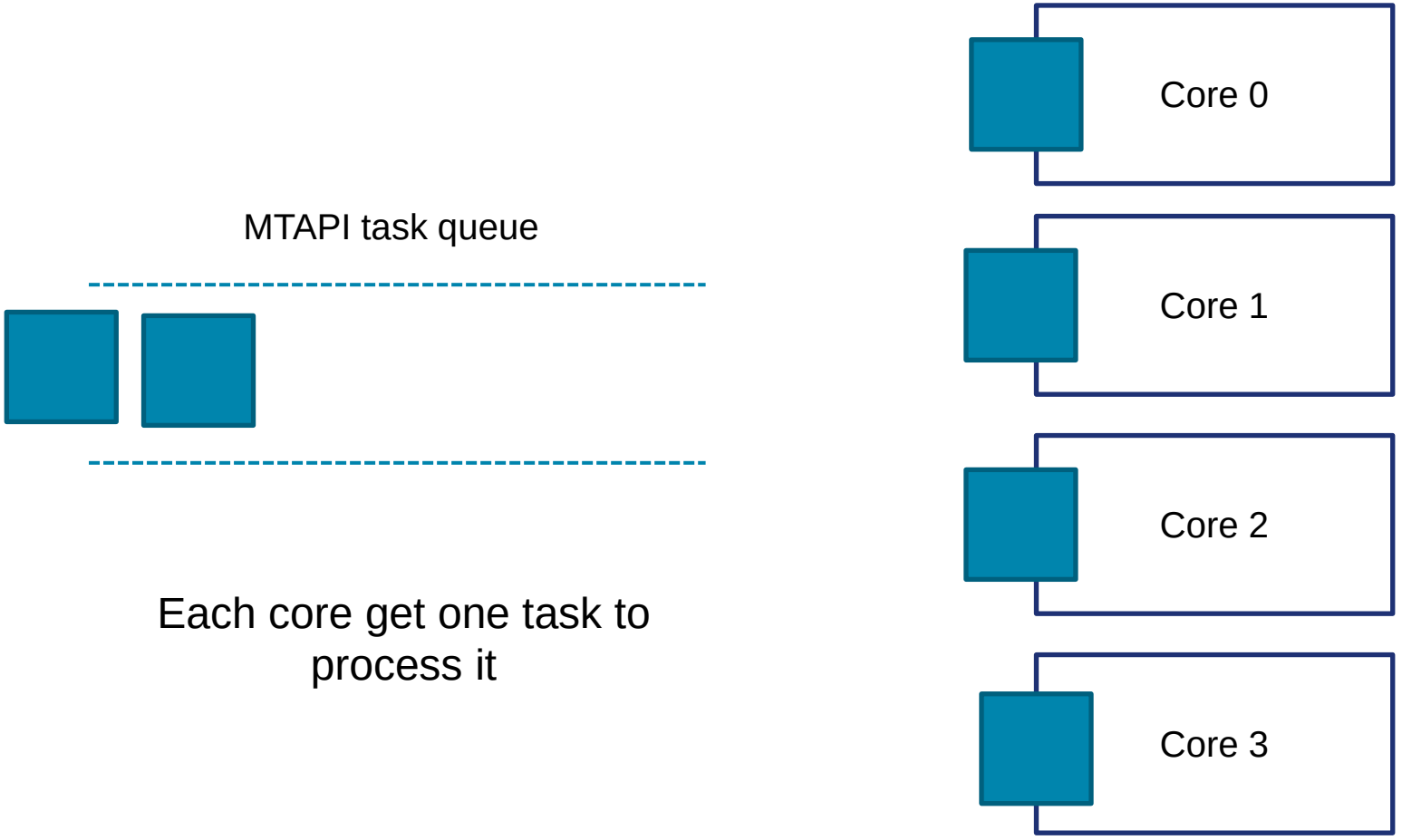


Tasks are stored by MTAPI



AIRBUS DEFENCE & SPACE

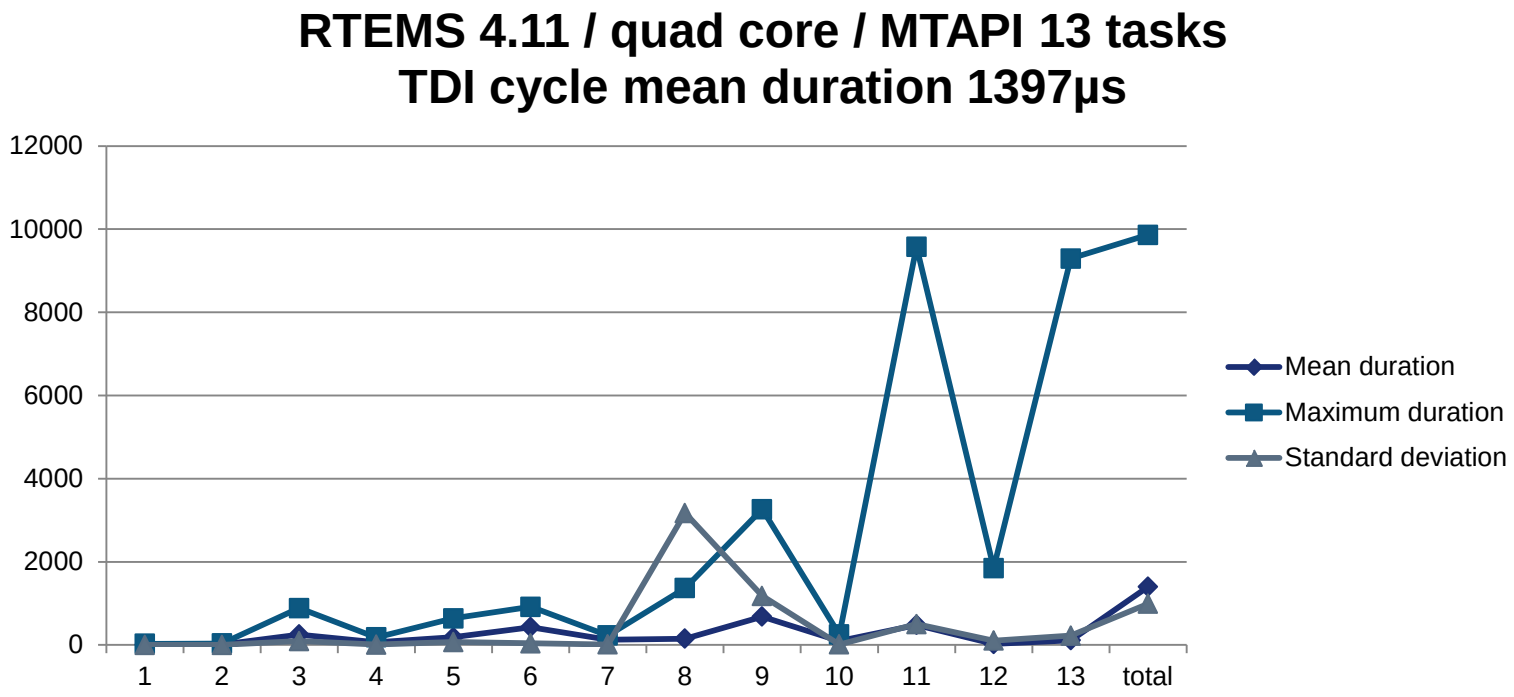
MTAPI Usage



Each core get one task to process it

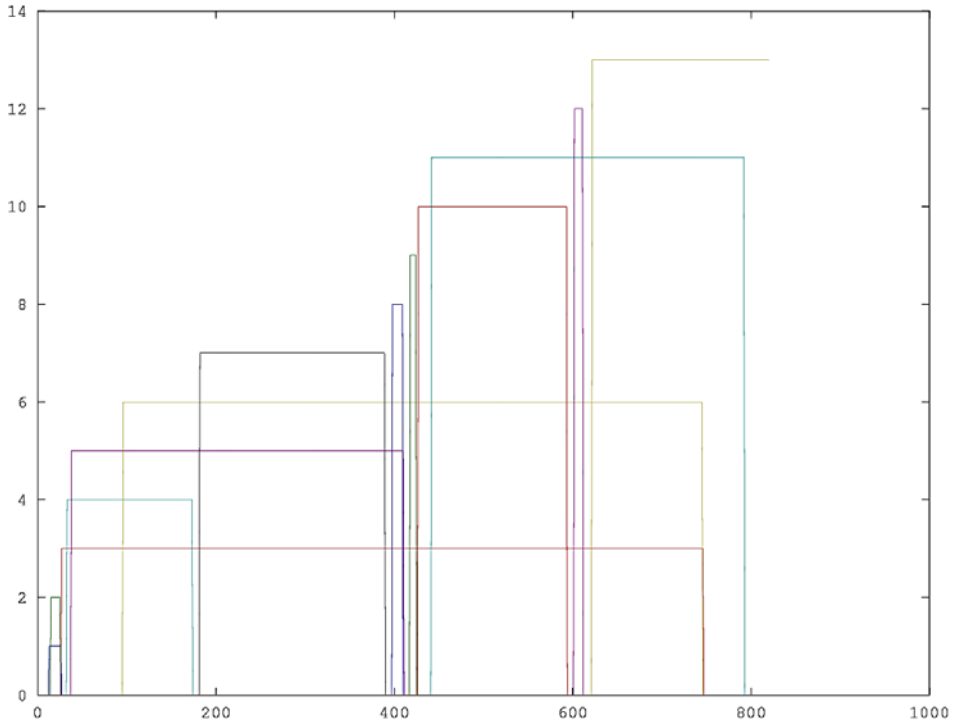
Measurements on NGFP @ 150 MHz

Performance improvement: x1.8

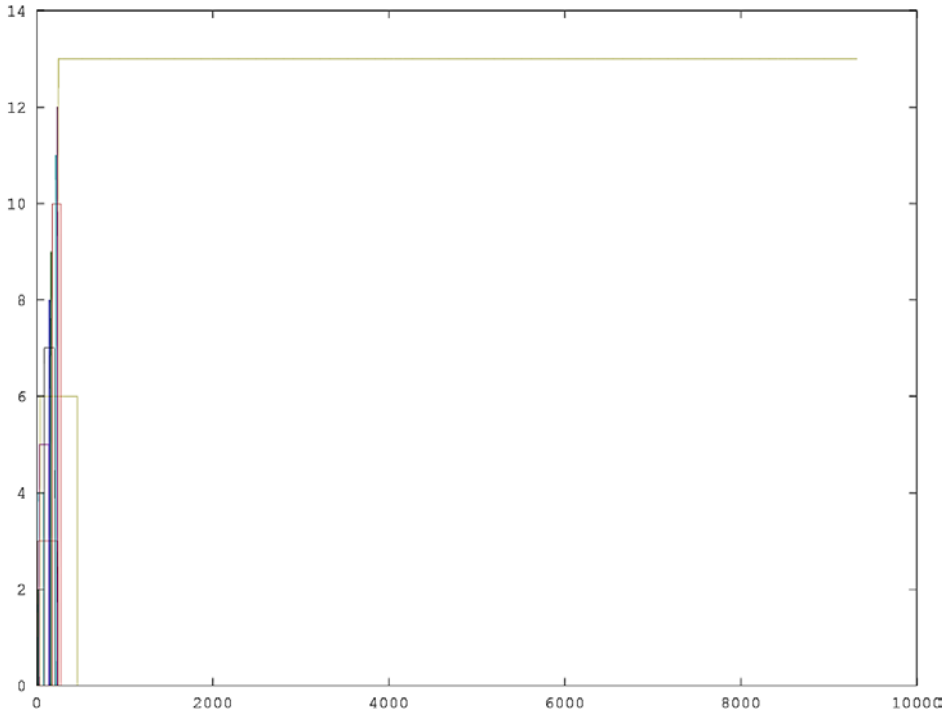


Parallelization scheme : Figure Of Merit

- **Figure Of Merit**
  - Defined to characterize how well the processing load is balanced among the cores
  - $FOM(cycle) = \frac{\sum task\ duration(cycle)}{total\ cycle\ duration(cycle)}$

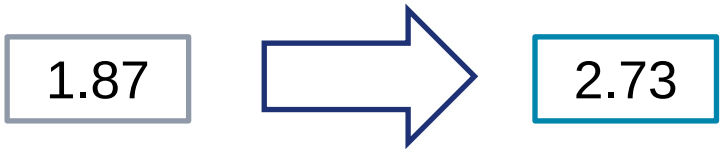


FOM = 3,42



FOM = 1,08

Load balancing optimizations

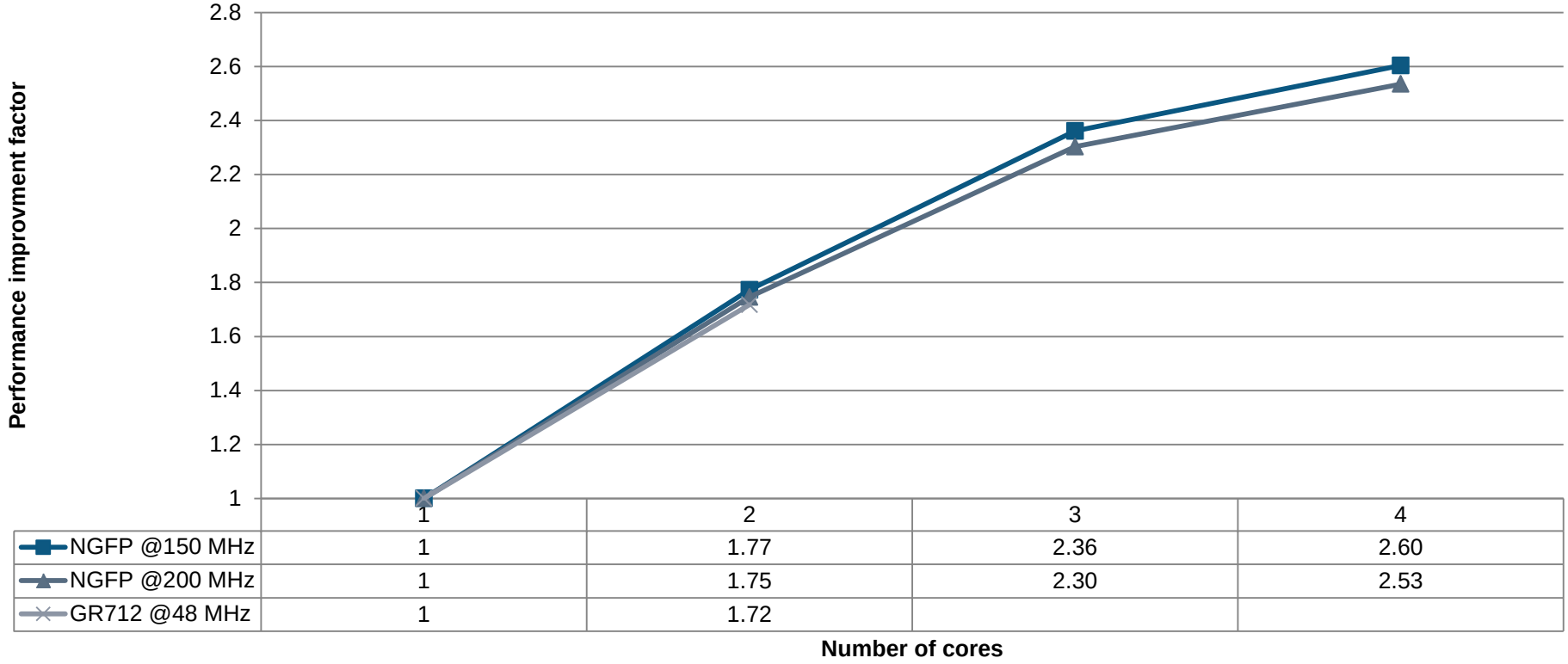


- **Tasks refactoring**
  - “Long” tasks are split in several tasks
  - Merge of short tasks
  - New architecture with 16 tasks
  - Small updates on accesses to shared resources to avoid long locking times

- **Tasks execution re-ordering**
  - Start long task execution first
  - Only based on statistical information / weak optimization due to task duration variation
  - Not obvious with current MTAPI specification : no semantics to define action execution priority order

AIRBUS DEFENCE & SPACE

Core scaling

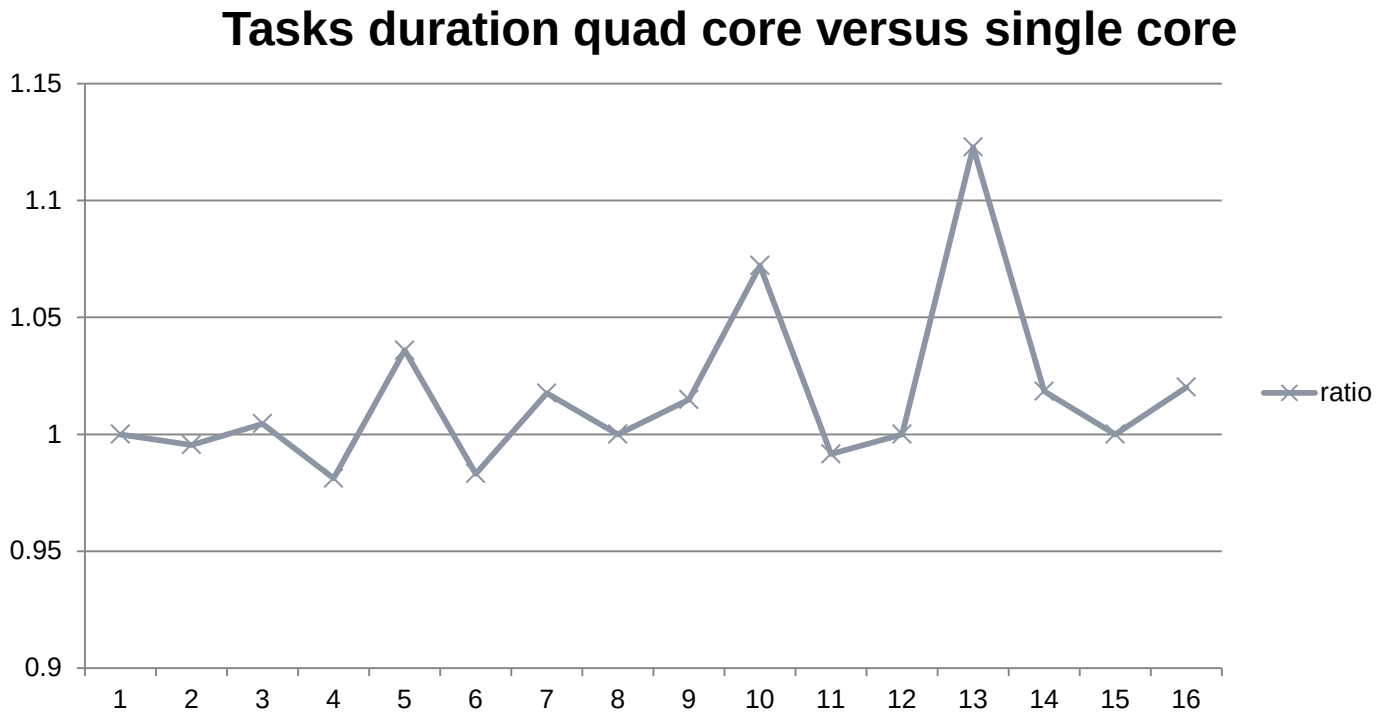


	1	2	3	4
NGFP @150 MHz	1	1.77	2.36	2.60
NGFP @200 MHz	1	1.75	2.30	2.53
GR712 @48 MHz	1	1.72		

- Scaling limitied by application parallelization (FOM = 2.7)
- Scaling on NGFP is better than GR712 , impact of L2 cache (96.5% cache hit ratio)

AIRBUS DEFENCE & SPACE

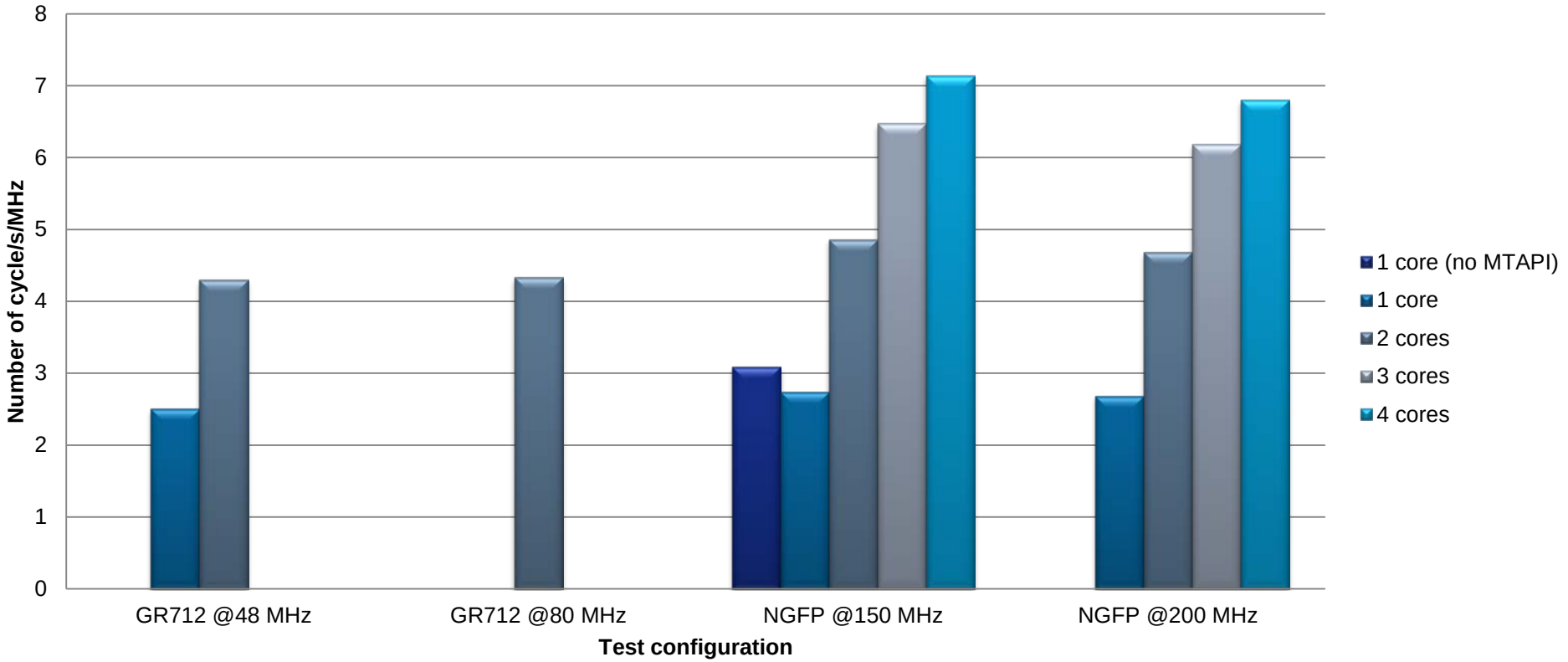
Concurrent task execution overhead measurement



- Measured with on a cycle with FOM = 3.81 (very good parallelization)
- Maximum intercore interference +12%
- Some tasks execute faster in concurrent setup (cache locality benefit)

AIRBUS DEFENCE & SPACE

Overall test results



- MTAPI overhead 12.5%
- Performance /MHz of NGFP down 4.7% due to memory/core ratio
- Performance of GR712 in dual core is 13% lower than NGFP (impact of memory bandwidth and inter core interference)

AIRBUS DEFENCE & SPACE

Return of experience

MTAPI

- Straightforward use
- No bugs discovered
- Only 1000 lines of code modified

Parallelization of the GAIA VPU application

- Straightforward parallelization thanks to initial application design (most of the functions are reentrant, application design already based on independent tasks)
- Finding and correcting remaining unprotected shared resources took some time and required in depth knowledge of the application
- Task duration variability limits efficient load balancing

Speed up of 2.6 from single core to 4 cores

➢ **Could be improved by further parallelization of the application**

GAIA VPU requirement: 982µs

➢ **Could be achieved with future GR740 @ 250 MHz**

AIRBUS DEFENCE & SPACE