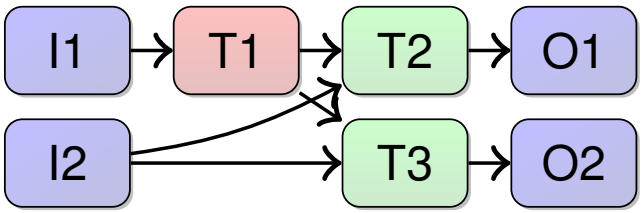
Response-Time Analysis of Dataflow Applications on a Many-Core Processor with Shared Memory and Network on Chip

Amaury Graillat, Claire Maiza, Matthieu Moy,
Pascal Raymond, Benoit Dinechin

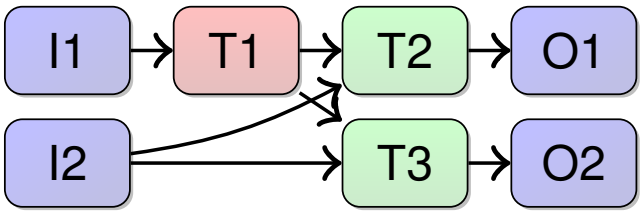
RTNS 2019

Our aim

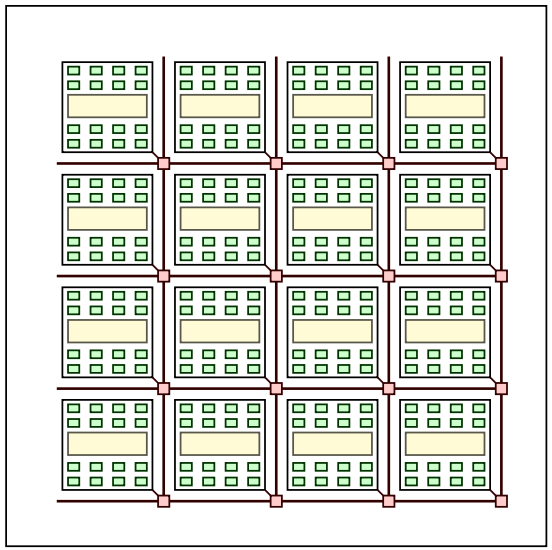


High-level Data-Flow Application

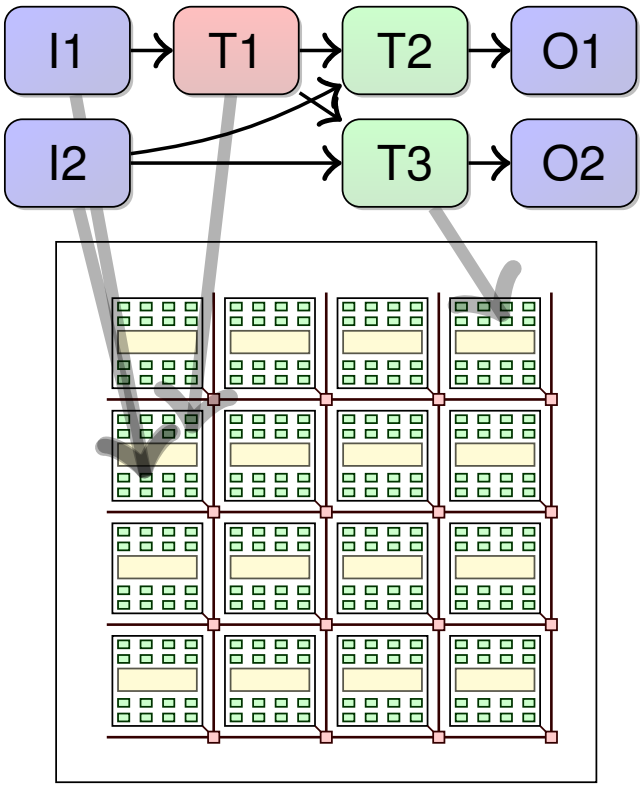
Our aim



High-level Data-Flow Application

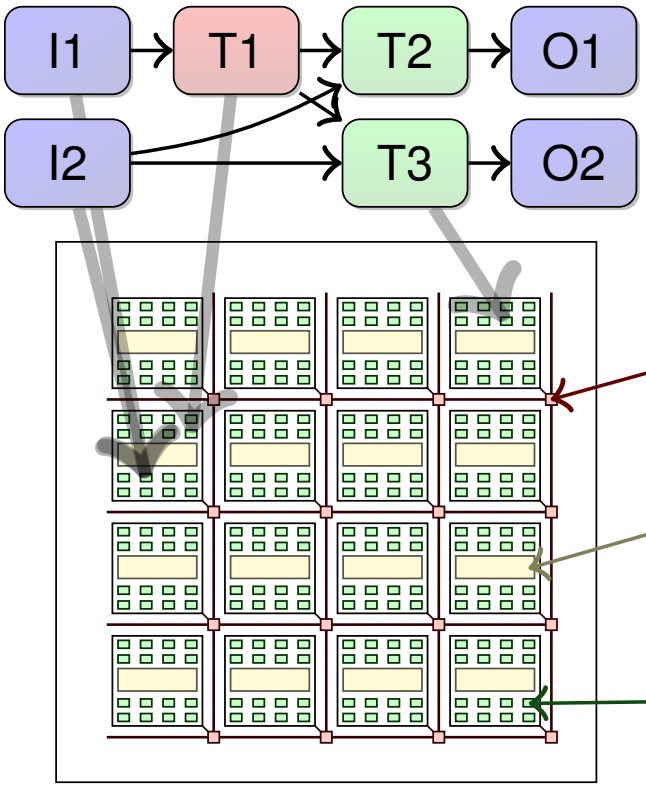


Our aim



High-level Data-Flow Application

Our aim



High-level Data-Flow Application

Network On Chip

Communication takes time

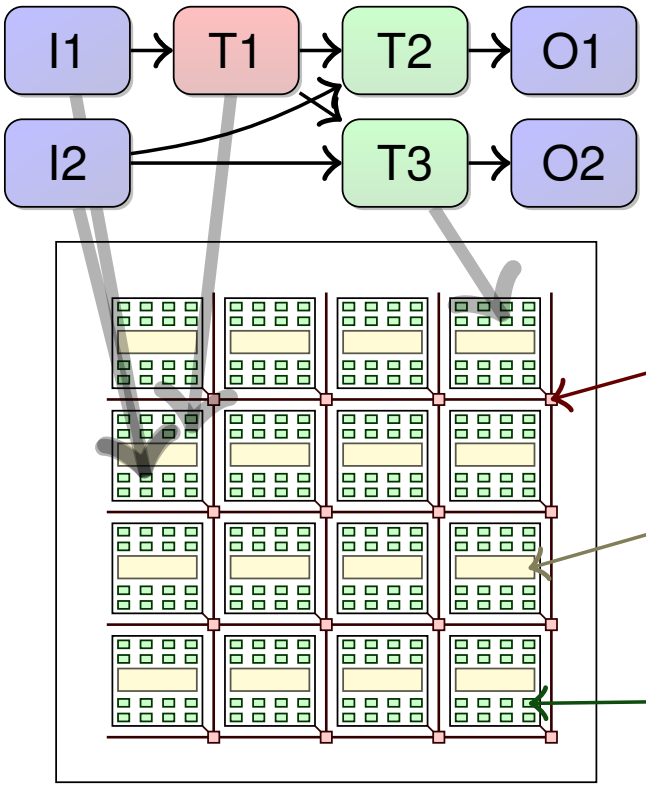
Shared Memory within Cluster

Interferences between tasks

Individual Cores

Cache, Pipeline, . . .

Our aim



High-level Data-Flow Application

Network On Chip

Communication takes time

Shared Memory within Cluster

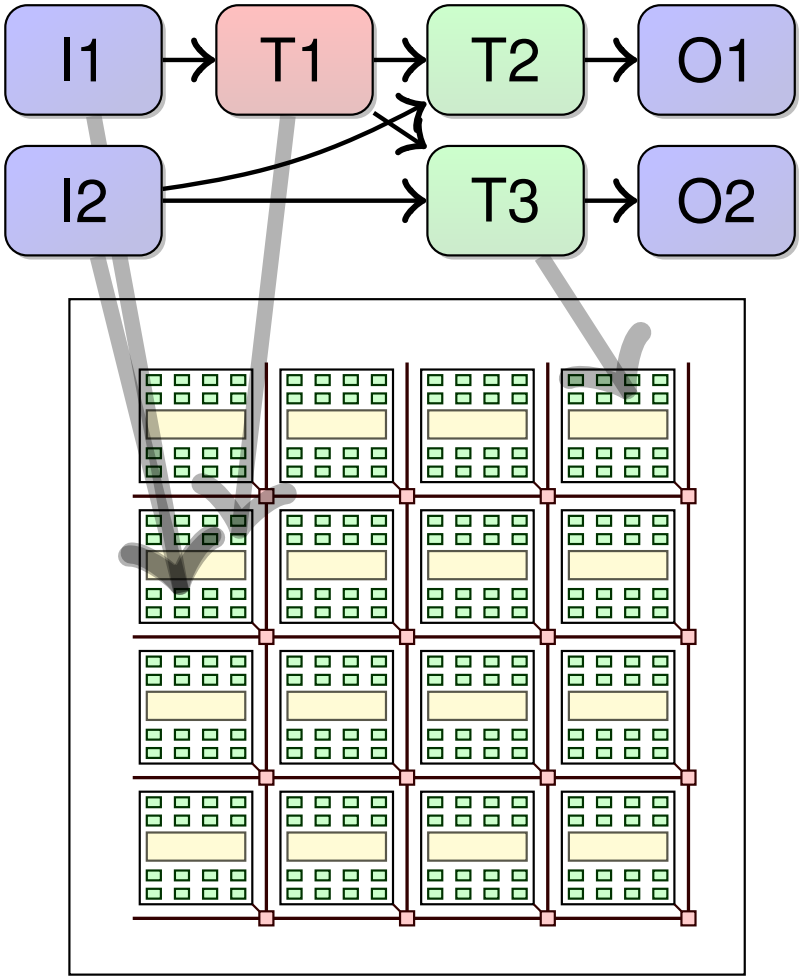
Interferences between tasks

Individual Cores

Cache, Pipeline, . . .

$\rightsquigarrow$ Implementation of a dataflow application on a many-core

$\rightsquigarrow$ with intra-cluster communications



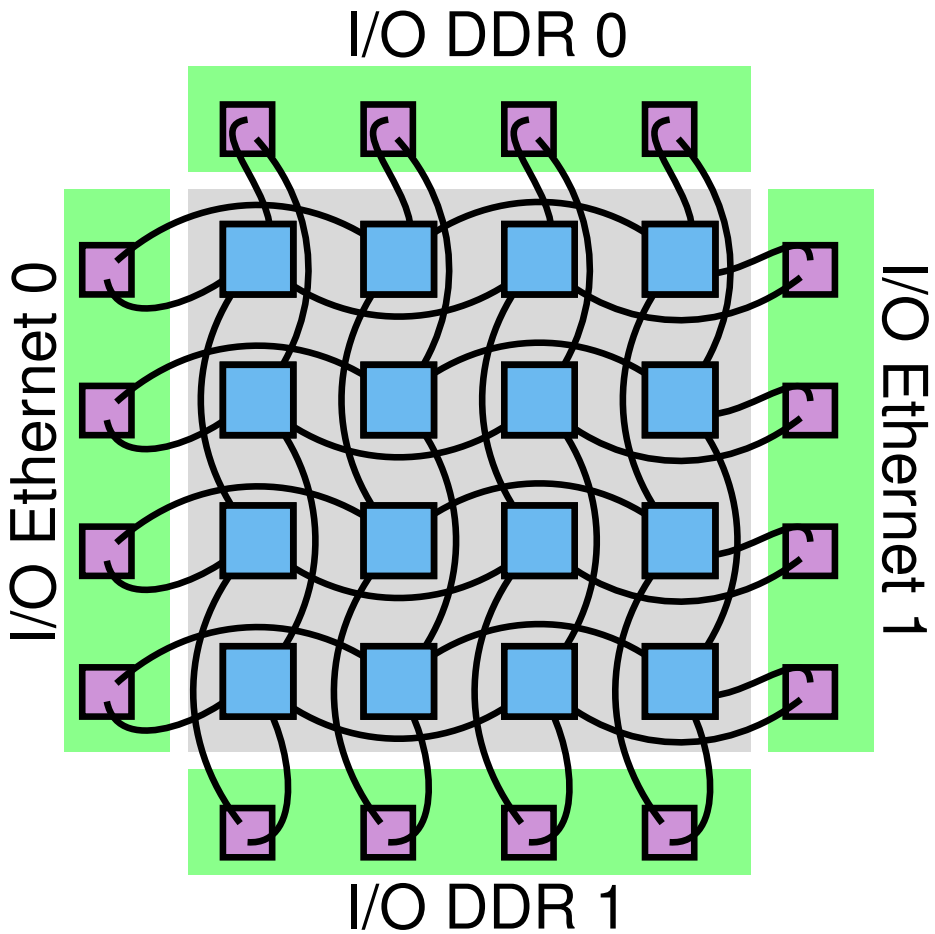
In this talk

- The basis: intra-cluster implementation and analysis
- The question : Inter-cluster, what are the main steps of a communication?
- Our timing analysis and implementation choices on the MPPA2 many-core

Outline

Outline

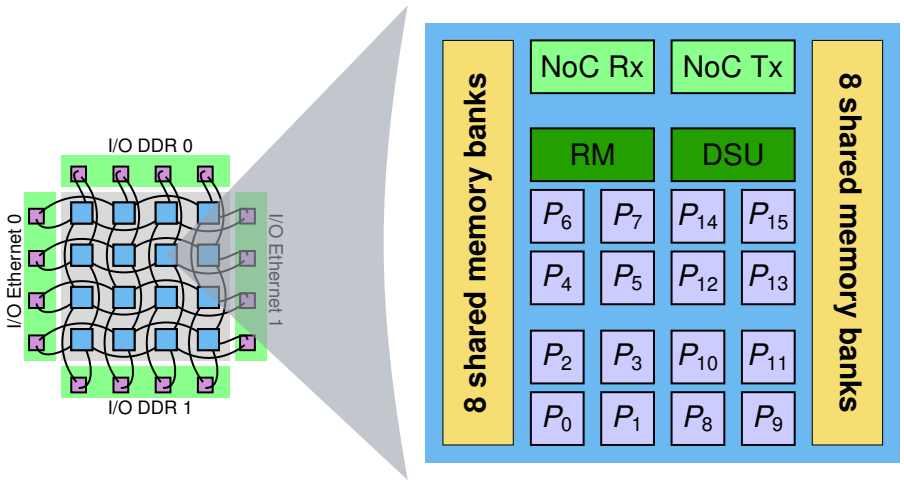
Architecture Model



Kalray MPPA 256 Bostan

16 compute clusters + 4 I/O clusters
Dual NoC (Network on Chip)

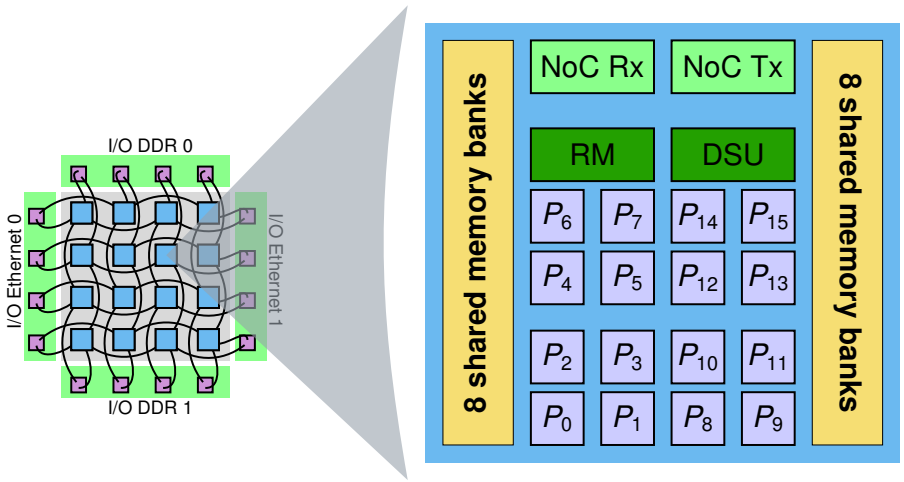
Architecture Model



Per cluster:

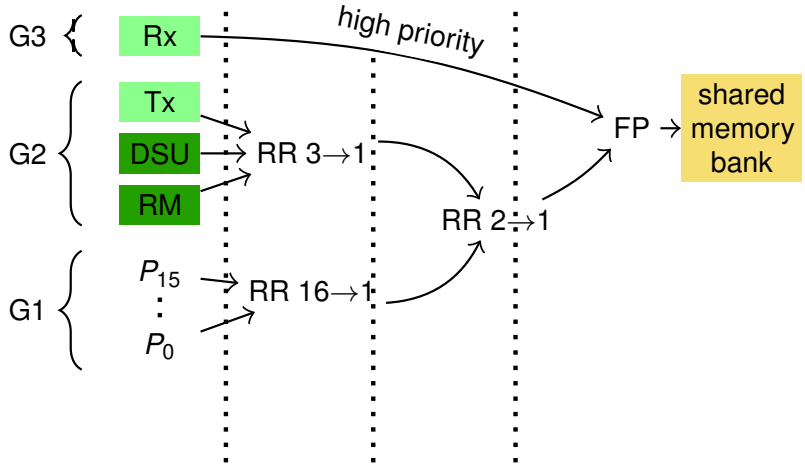
16 cores + 1 Resource Manager
NoC Tx, NoC Rx, Debug Unit
16 shared memory banks
(total size: 2 MB)
Multi-level bus arbiter per memory bank

Architecture Model

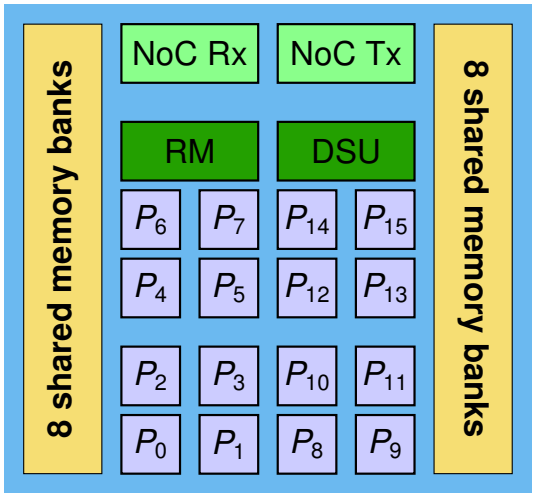


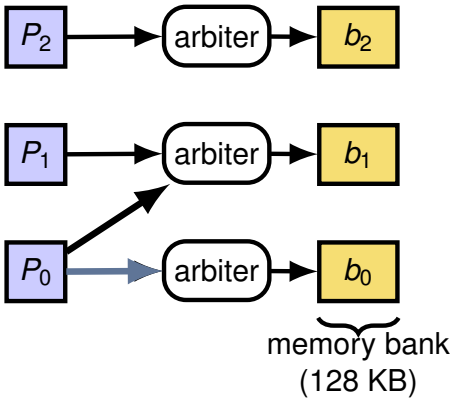
Per cluster:

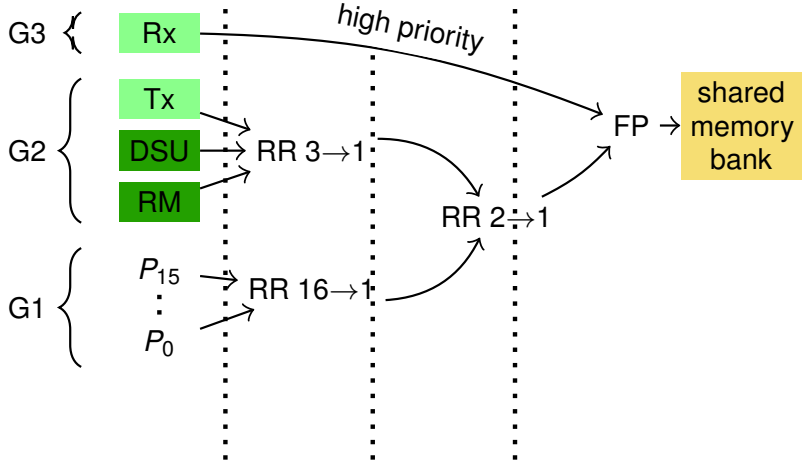
16 cores + 1 Resource Manager
NoC Tx, NoC Rx, Debug Unit
16 shared memory banks
(total size: 2 MB)
Multi-level bus arbiter per memory bank



Memory mapping and execution model







Implementation choice

Phased execution model:

- Execute in a “local” bank
- Write to a “remote” bank

$\Rightarrow$ Interference limited to communication (write)

Response Time Analysis : within a cluster

Response Time Analysis[1]

$$R_i = WCET_i + \sum_{j \neq i} interference_{i,j}$$

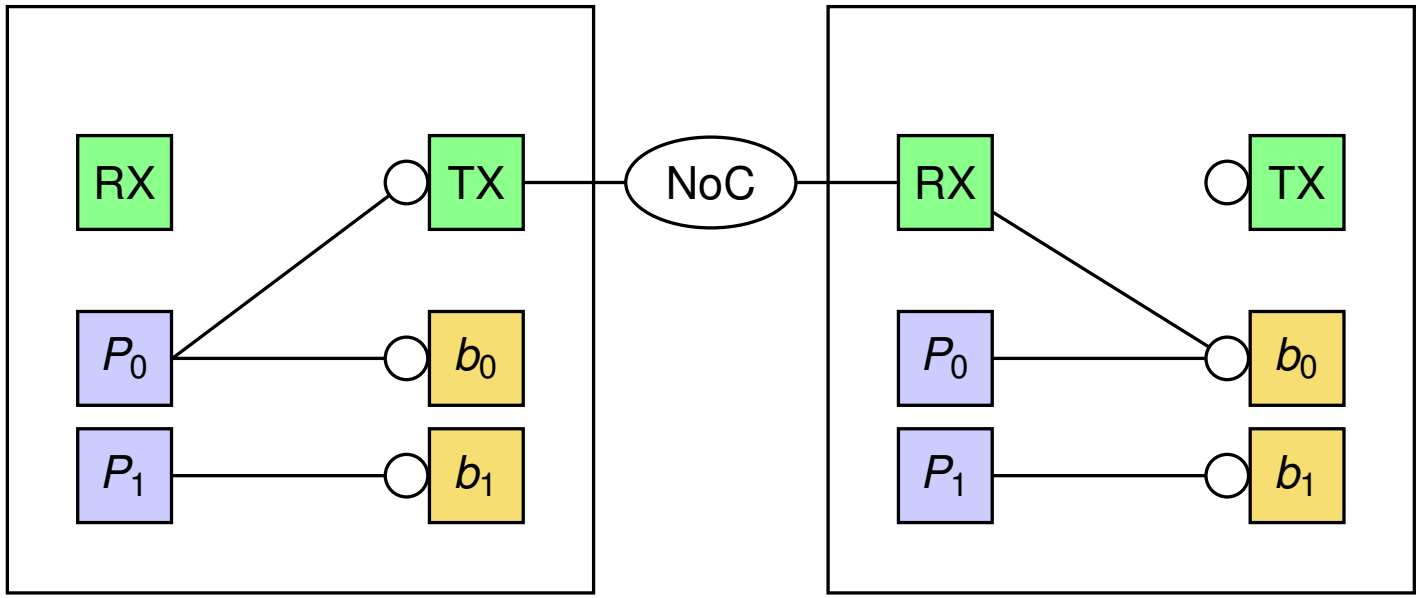
MIA tool: Interference Analysis, Response Time and Release date

- From: local WCET, # memory accesses, initial Scheduling/Mapping
- Estimate interference delay
- Reajust release date to respect precedence constraints
- Iterate until fix point

[1] no preemption

Outline

NoC Communication in MPPA2



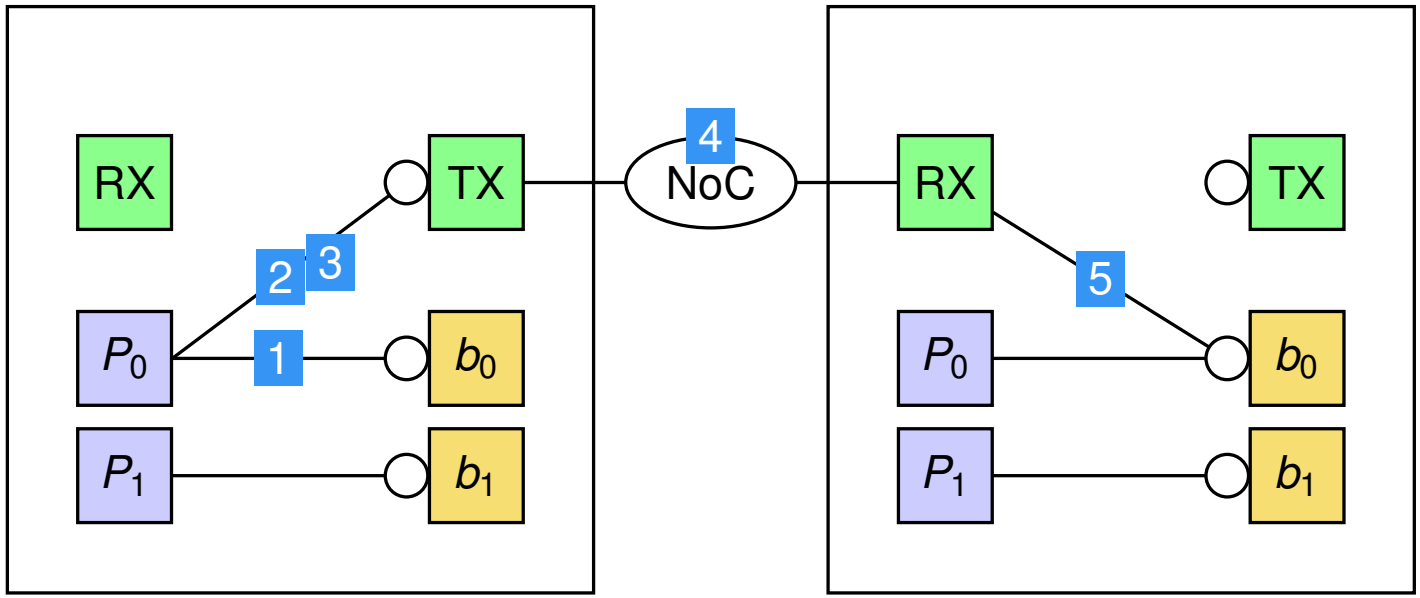
NoC Communication steps:

1. Read from bank

Interference:

1. Intra-cluster bus interference

NoC Communication in MPPA2



NoC Communication steps:

1 Read from bank

2, 3 write to the buffer and start transmission

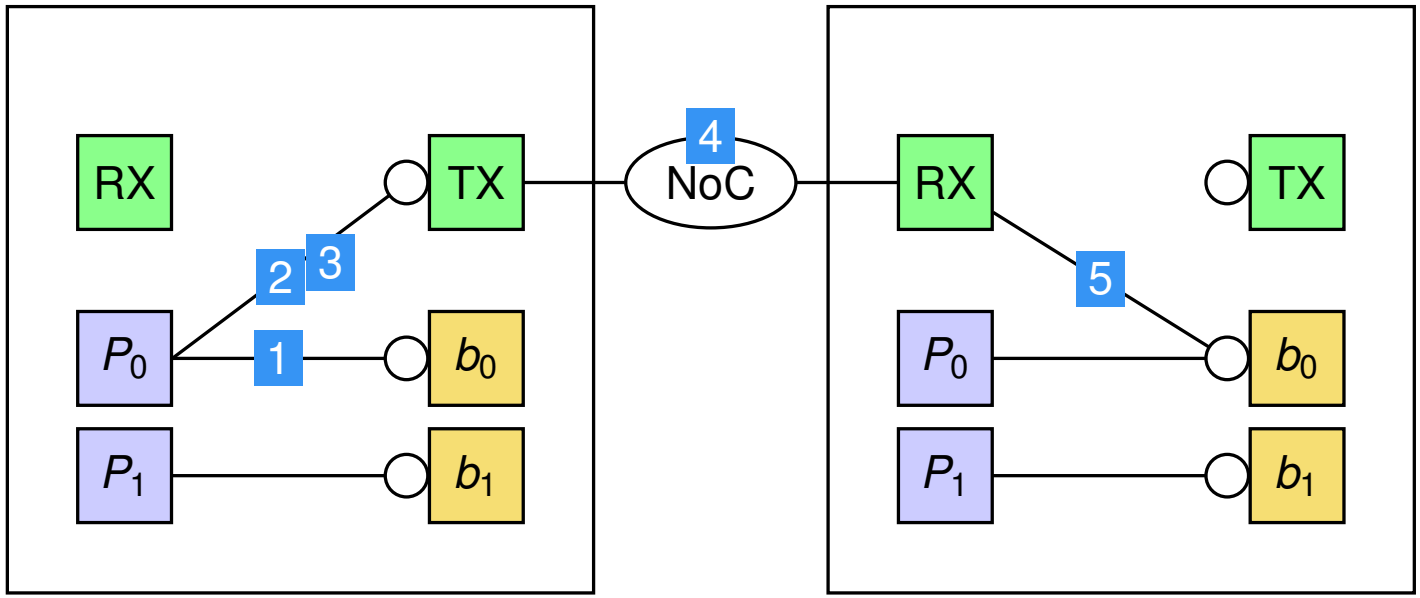
Interference:

1 Intra-cluster bus interference

2, 3 no arbiter $\Rightarrow$ One TX channel per sender

$\Rightarrow$ isolation

NoC Communication in MPPA2



NoC Communication steps:

1 Read from bank

2, 3 write to the buffer and start transmission

4 NoC transmission

Interference:

1 Intra-cluster bus interference

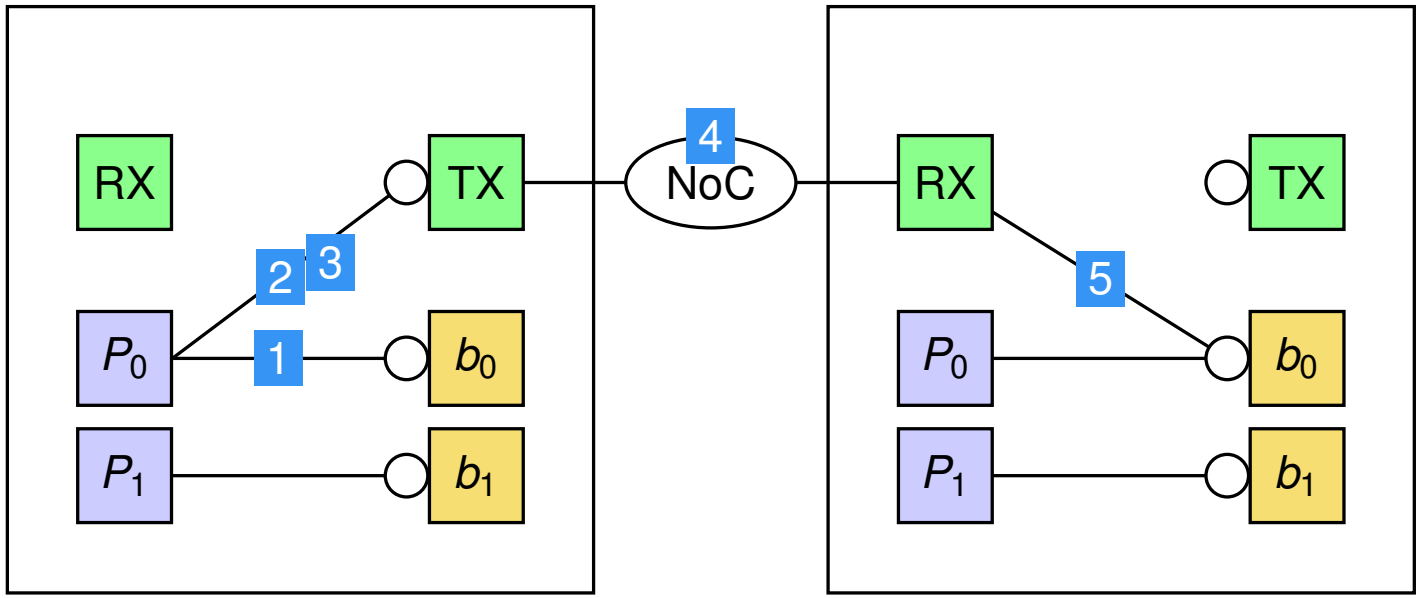
2, 3 no arbiter $\Rightarrow$ One TX channel per sender

$\Rightarrow$ isolation

4 Interferences in each router

$\rightarrow$ network calculus

NoC Communication in MPPA2



NoC Communication steps:

1 Read from bank

2, 3 write to the buffer and start transmission

4 NoC transmission

5 write in bank with High-priority!

Interference:

1 Intra-cluster bus interference

2, 3 no arbiter $\Rightarrow$ One TX channel per sender

$\Rightarrow$ isolation

4 Interferences in each router

$\rightarrow$ network calculus

5 Need of interference analysis

Outline

How to integrate the NoC Reception in the timing model?

NoC reception as an additional task

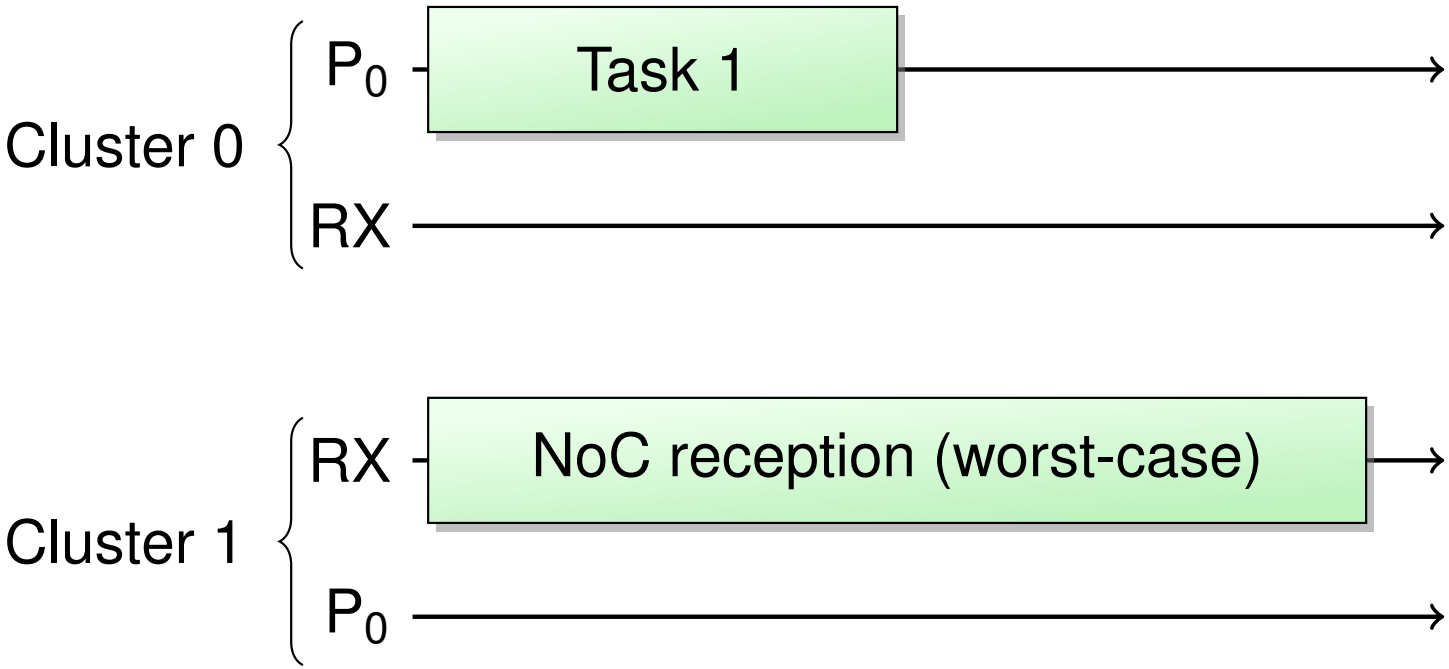
- Model as a task $\Rightarrow$ when the fixed priority interference will occur

Q1 When does the reception start?

Q2 What about circular dependency?

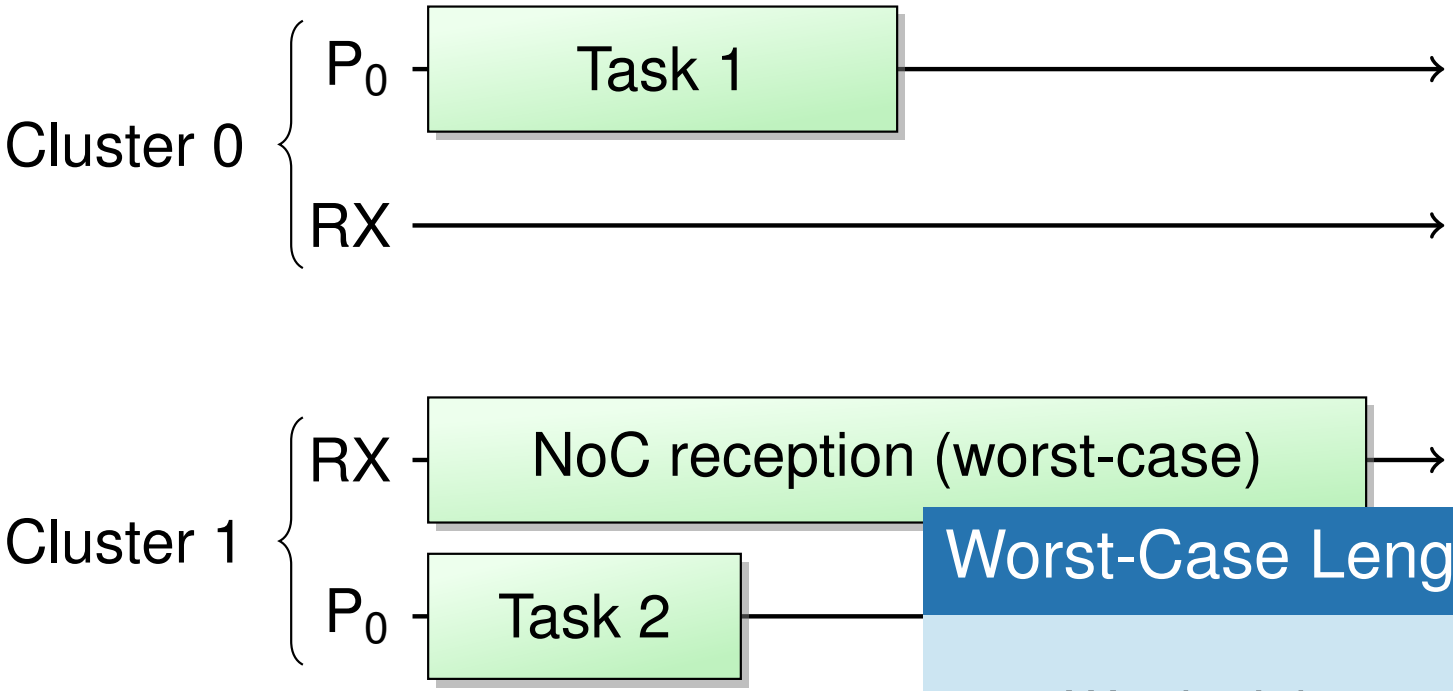
Example Tasks with NoC Transmission

Question 1: How do we model the NoC transmission?



Example Tasks with NoC Transmission

Question 1: How do we model the NoC transmission?



Worst-Case Length

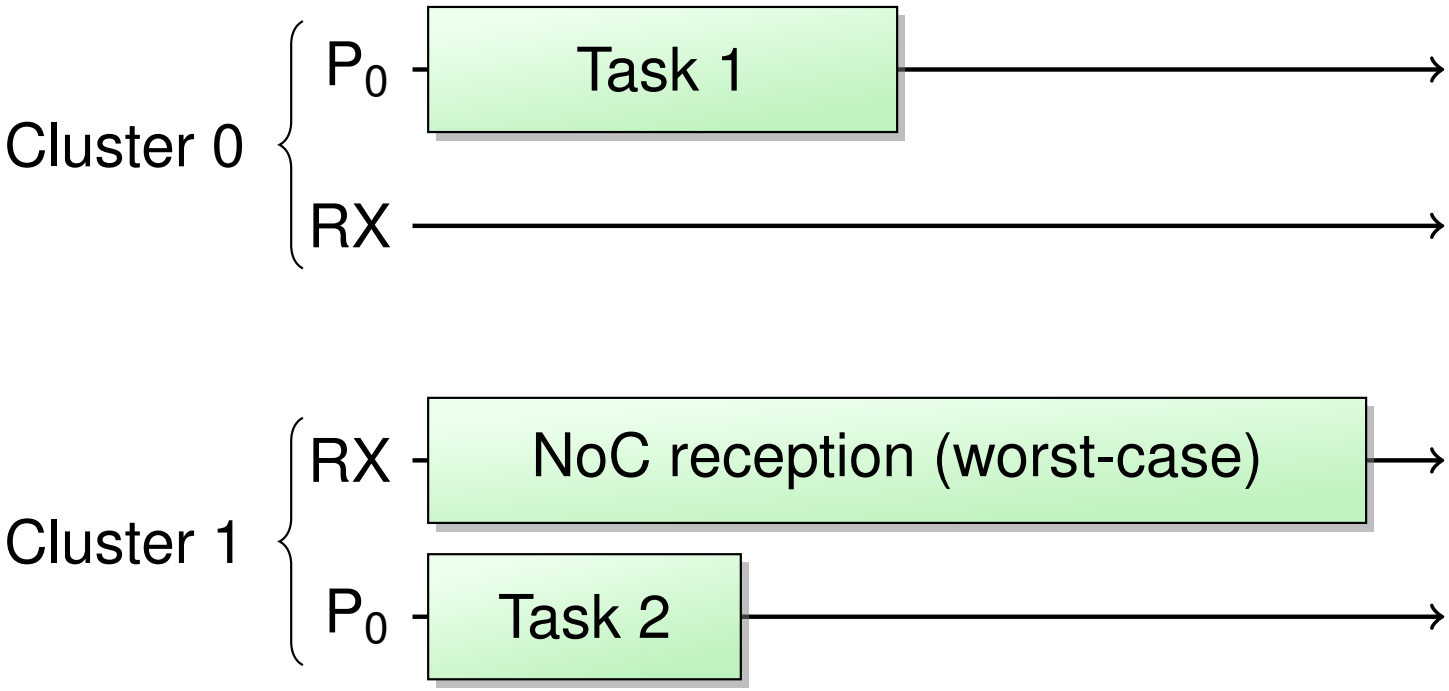
- We don't know when the NoC transmission happens

$\Rightarrow$ Worst-case = from the first cycle of Task 1!

- Reception = WCRT of sending task + WCTT NoC + Writing time in Cluster memory

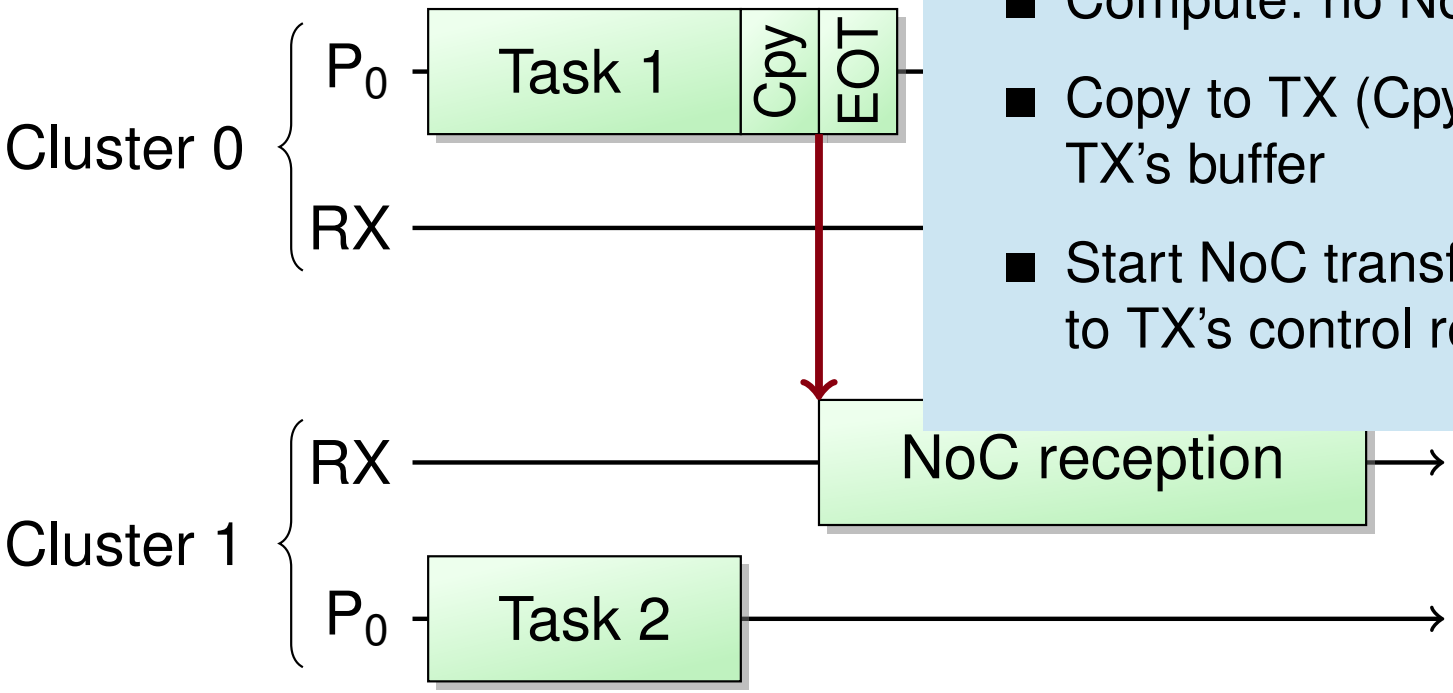
Example Tasks with NoC Transmission

Question 1: How do we model the NoC transmission?



Example Tasks with NoC Transmission

Question 1: How do we model the NoC transmission?

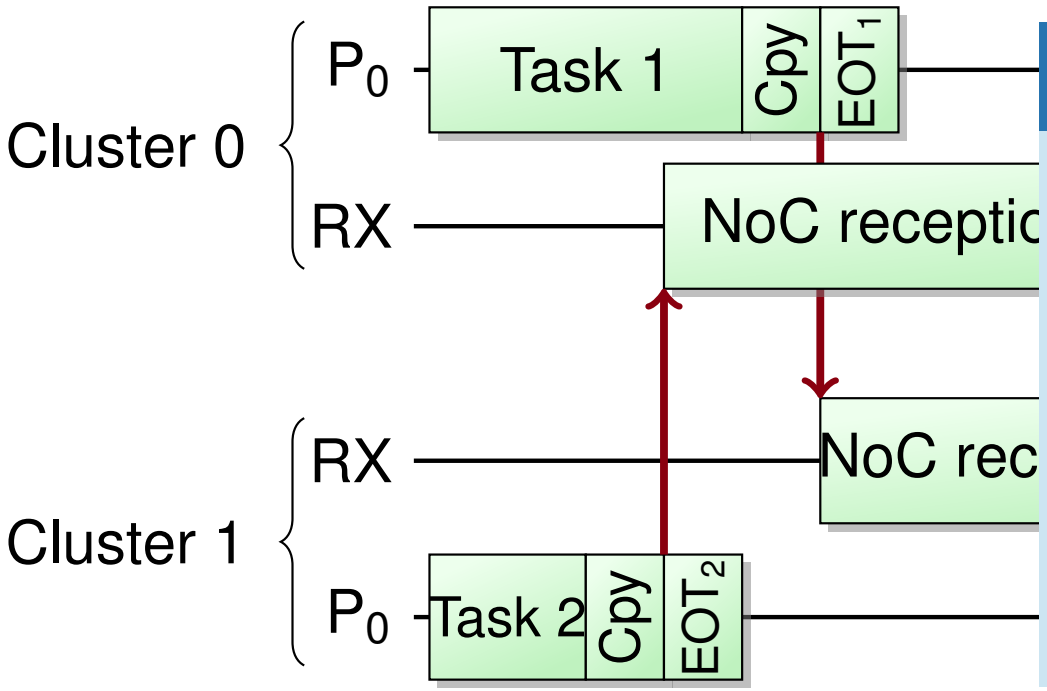


Better: Split sending task

- Compute: no NoC access
- Copy to TX (Cpy): write to the TX's buffer
- Start NoC transfer (EOT): write to TX's control register

Example Tasks with NoC Transmission

Issue: circular dependency?



Circular dependency

- EOT = one control register access
- Avoid memory access interference
  $\Rightarrow$ Preload code to avoid instruction cache miss

Summary: NoC Communication Timing Analysis

- Compute Task:
  - Fits in previous work memory interference model
- Copy to TX:
  - Isolate by scheduling
- Start NoC transfer (EOT):
  - Avoid memory access = no interference
- On the RX side:
  - RX can only start after “Start NoC transfer”
    $\Rightarrow$ edge from “Copy to TX” to “RX” in the task dependency graph.

Outline

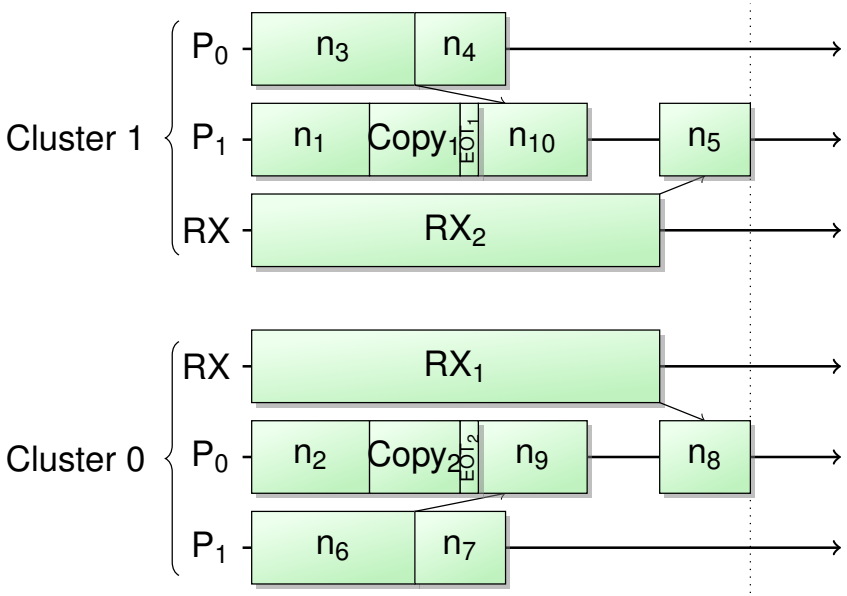
How to validate such a model?

Evaluated in the paper

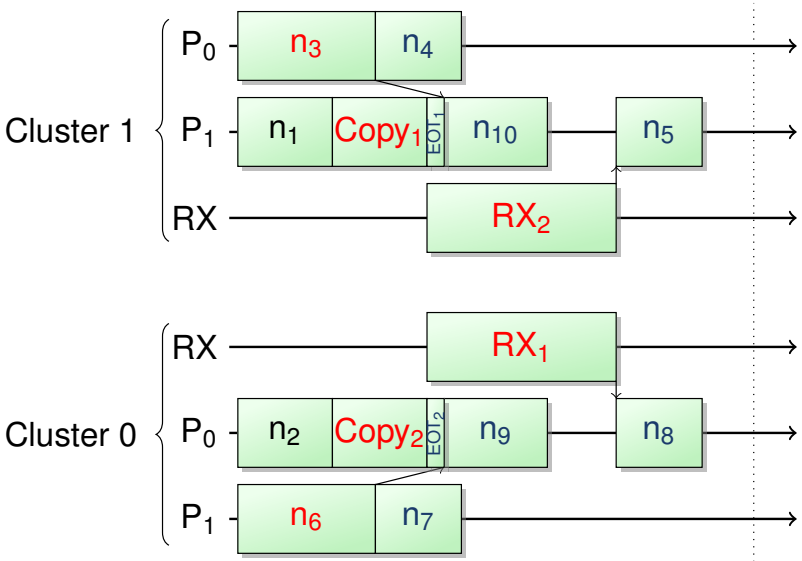
- How much do we gain on the Response-Time with our 3-phased Noc Transmission task?
- How precise is the time-triggered implementation?

Example Application

Worst-Case:



Improved:



Improved WCRT Earlier release date only

New Data-flow Application implementation and timing analysis with intra-cluster communications

Summary

- Sending task = 3-phased task
- Isolated NoC transmission
- No memory access during EOT task
- Interference due to NoC recption taken into account in Multi-Core Interference Analysis