Quentin Dariol’s PhD defense – 27.11.2023

Early Timing and Energy Prediction and Optimization of Artificial Neural Networks on Multi-Core Platforms

Evaluation committee:

- Dr. Kim GRÜTTNER
- Prof. Dr. Matthias JUNG
- Dr. Angeliki KRITIKAKOU
- Prof. Dr. Frédéric PÉTROT
- Prof. Dr. Gregor SCHIELE

Guest:

- Dr. Domenik HELMS

Reviewers before defense:

- Prof. Dr. Matthias JUNG
- Dr. Angeliki KRITIKAKOU

Thesis work supervised by:

- Prof. Dr. Sébastien PILLEMENT
- Dr. Sébastien LE NOURS

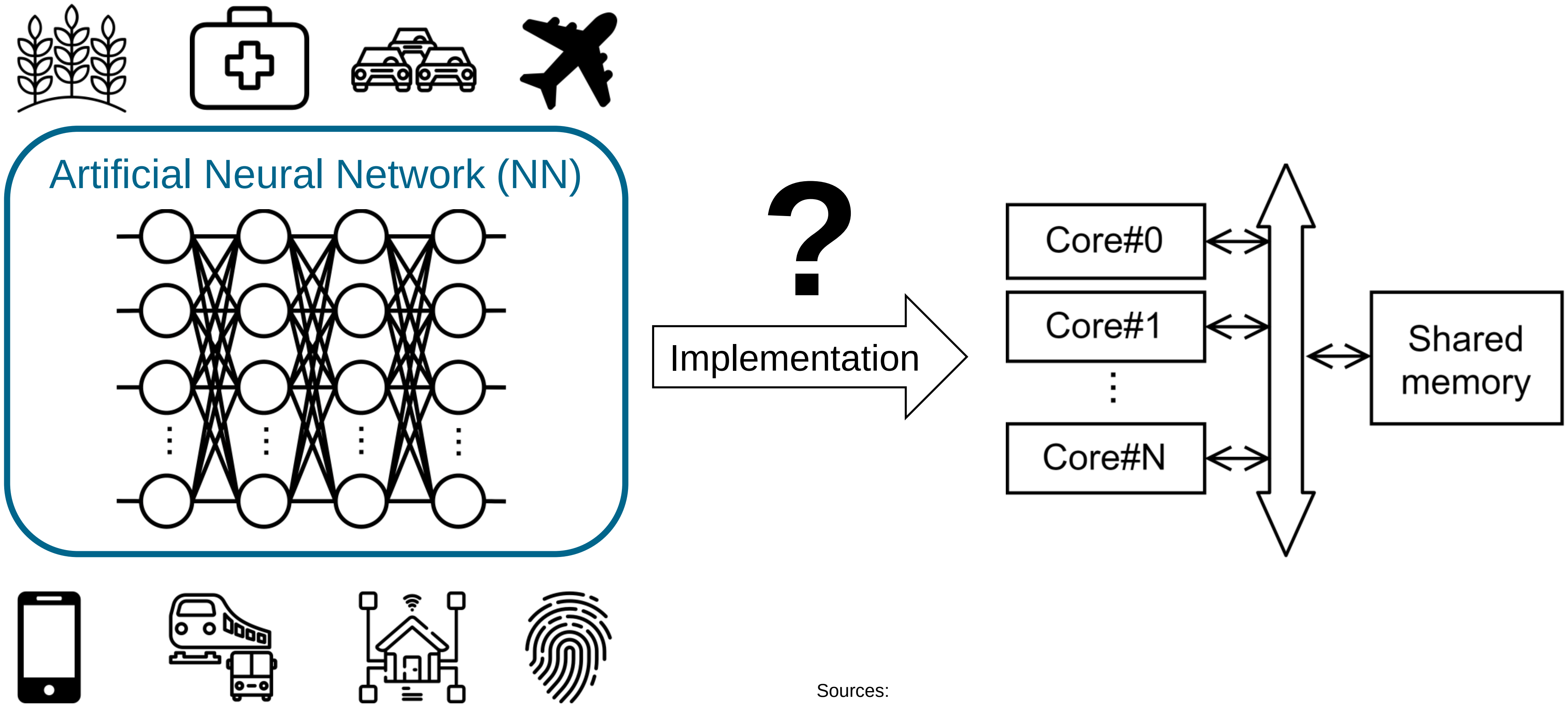
IETR

Nantes Université

DLR

Context – Artificial Neural Networks (NNs)

- Raise of interest for AI algorithms and especially for NNs.



Artificial Neural Network (NN)

?

Implementation

Core#0

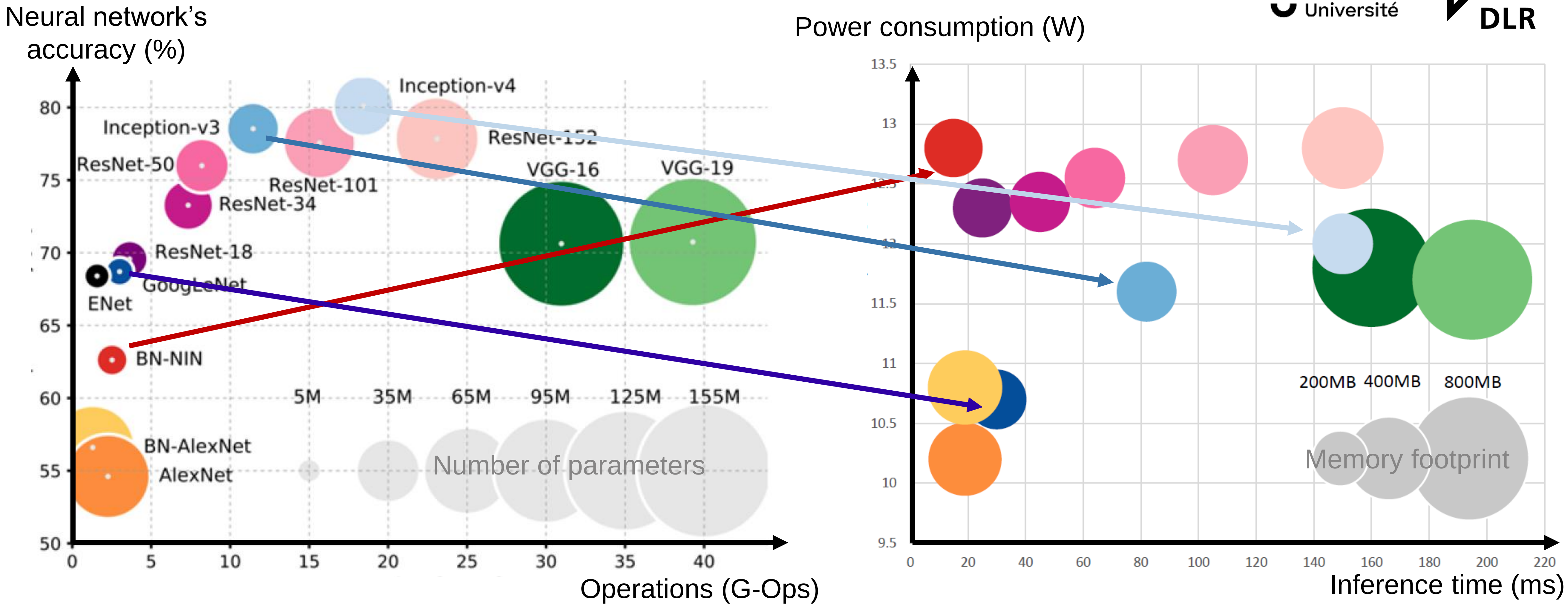
Core#1

Core#N

Shared memory

Sources:
Maslej, N.; et al., "The AI Index 2023 Annual Report", *Stanford Institute for Human-Centered Artificial Intelligence (HAI)*, 2022
Symbols from: https://www.flaticon.com/

Context – NNs on edge devices

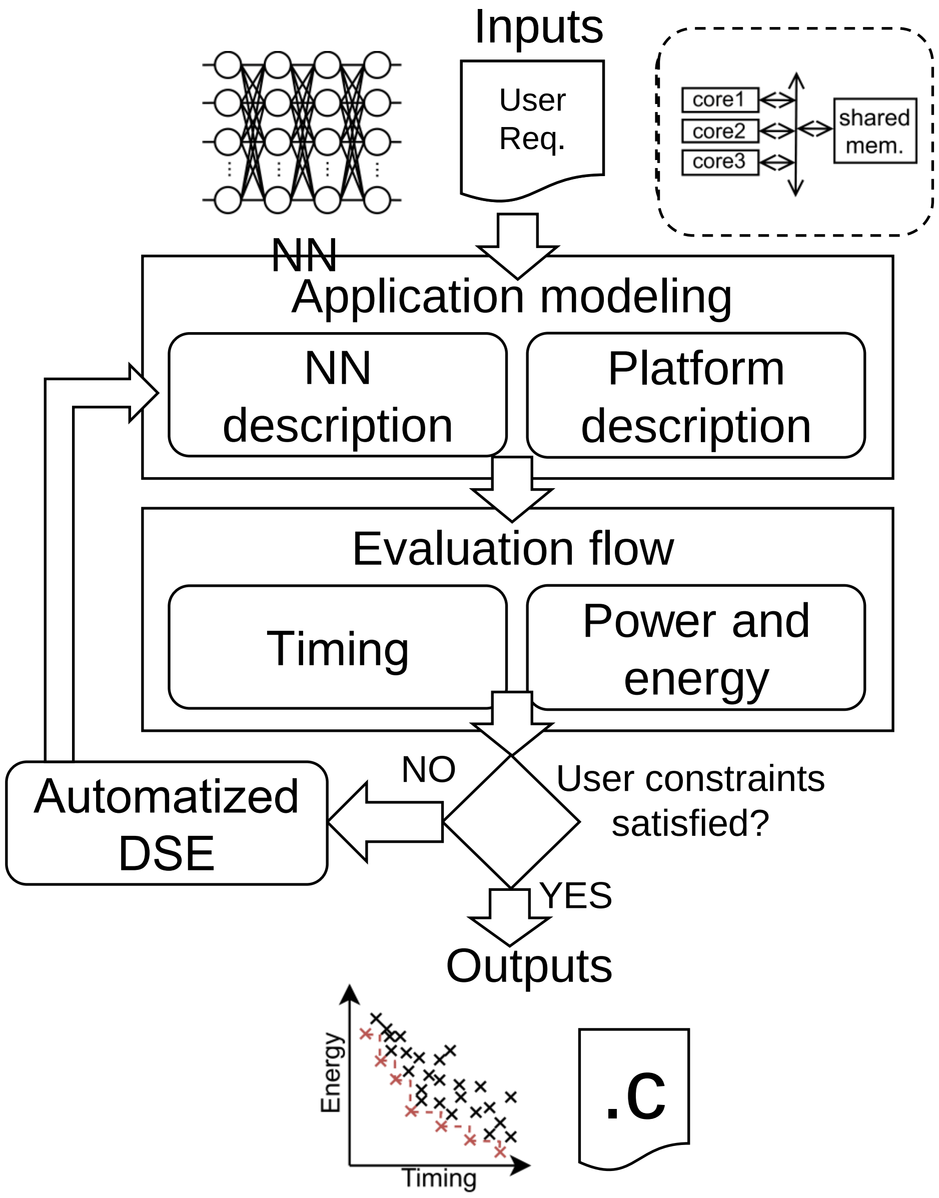


=> Metrics that matter at the edge

=> Need evaluation flow to find optimized mappings

Source: Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. *An Analysis of Deep Neural Network Models for Practical Applications*. 2017. arXiv: 1605.07678.

Proposition and presentation outline



I. Fundamentals & hypothesis

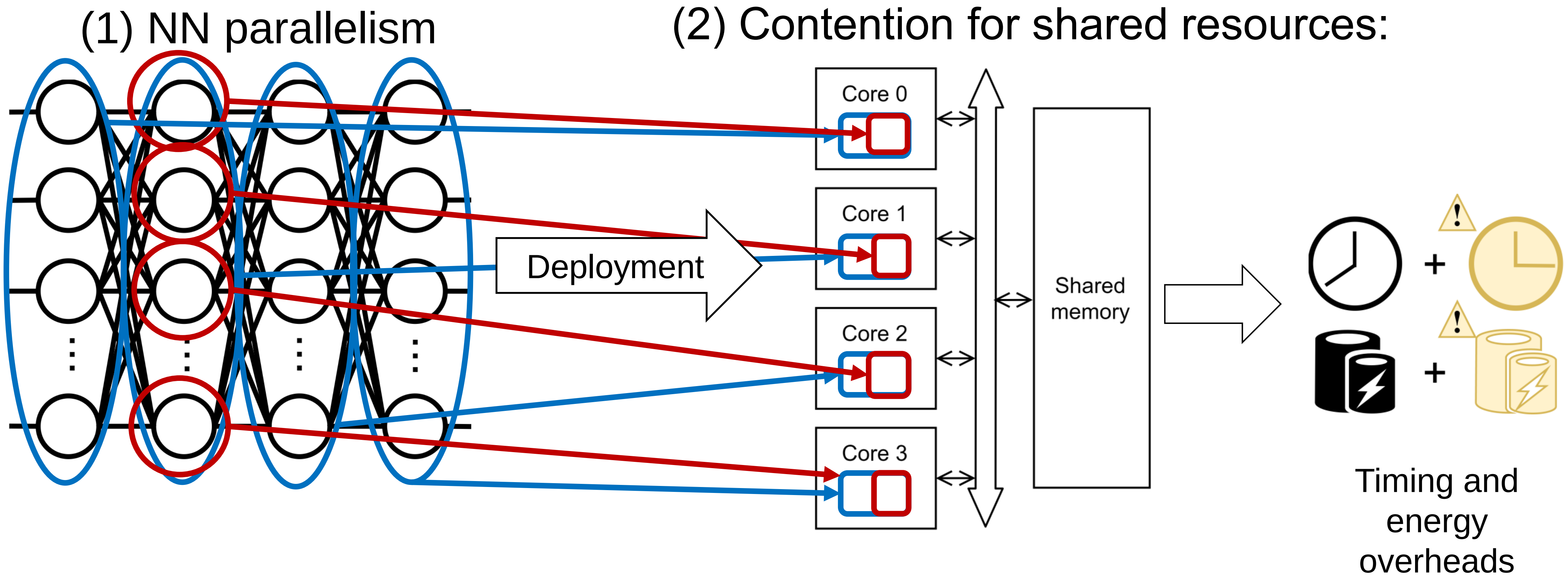
II. Timing prediction flow

III. Power and energy analysis flow

IV. Design Space Exploration (DSE) flow

V. Conclusion & Prospects

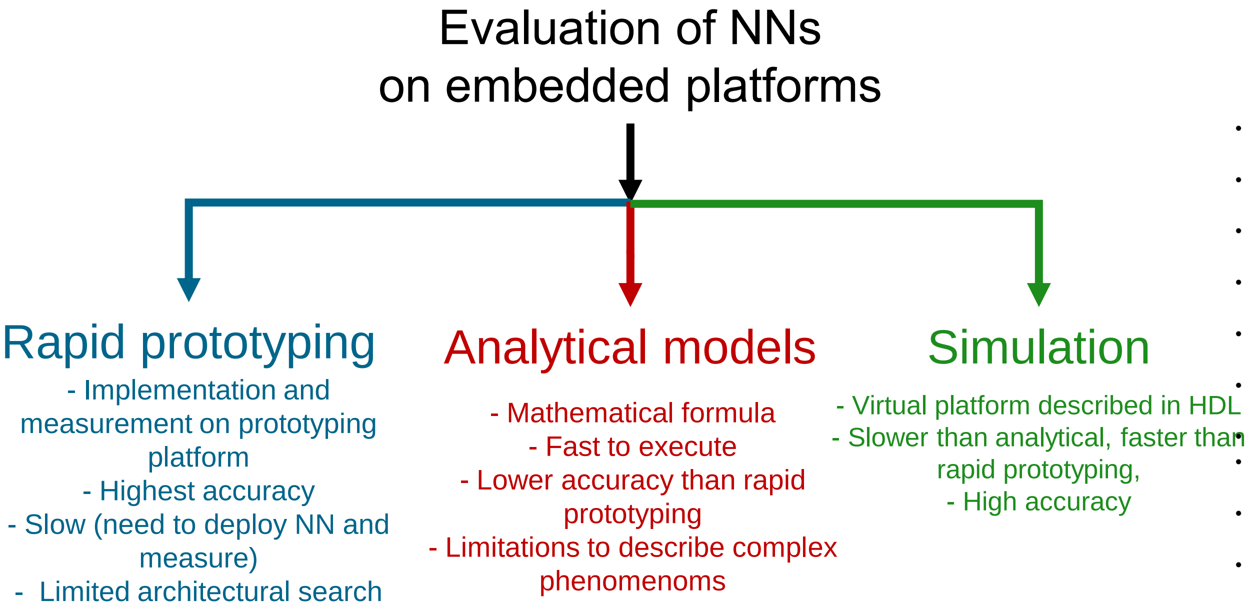
I. Fundamentals & hypothesis – Challenges of NN deployment on multi-core platforms



Other aspects:

- Use of power management
- Platform size (number of cores, memory)
- NN different workloads => no « one fits all » solution

I. Fundamentals & hypothesis – Related work



Evaluation of NNs on embedded platforms

Rapid prototyping

- Implementation and measurement on prototyping platform
- Highest accuracy
- Slow (need to deploy NN and measure)
- Limited architectural search

Analytical models

- Mathematical formula
- Fast to execute
- Lower accuracy than rapid prototyping
- Limitations to describe complex phenomenoms

Simulation

- Virtual platform described in HDL
- Slower than analytical, faster than rapid prototyping,
- High accuracy

- [Galanis2020] Galanis I. et al. "Inference and Energy Efficient Design of Deep Neural Networks for Embedded Devices", IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2020
- [Tsimpourlas2018] Tsimpourlas F. et al. "A Design Space Exploration Framework for Convolutional Neural Networks Implemented on Edge Devices", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCADICS), 2018
- [VelascoMontero2020] Velasco Montero D. et al. "PreVIous: A Methodology for Prediction of Visual Inference Performance on IoT Devices", IEEE Journal of Internet of Things, 2020
- [Guo2023] Guo X. et al. "Automated Exploration and Implementation of Distributed CNN Inference at the Edge", IEEE Journal of Internet of Things, 2023
- [Osterwind2022] Osterwind A. et al. "Hardware Execution Time Prediction for Neural Network Layers", IoT, Edge, and Mobile for Embedded Machine Learning (ITEM), 2022
- [Venieris2019] Venieris, S. and Bouganis, C.-S. "fpgaConvNet: Mapping Regular and Irregular Convolutional Neural Networks on FPGAs", IEEE Transactions on Neural Networks and Learning Systems, 2019
- [Parashar2019] Parashar, A. et al. "Timeloop: A Systematic Approach to DNN Accelerator Evaluation", ISPASS 2019
- [Garbay2021] Garbay, T. et al. "CNN Inference Costs Estimation on Microcontrollers: the EST Primitive-based Model", IEEE International Conference on Electronics, Circuits, and Systems (ICECS), 2021
- [Lee2022] Lee, J. et al. "Implication of Optimizing NPU Dataflows on Neural Architecture Search for Mobile Devices" - ACM Transactions on Design Automation of Electronic Systems (TODAES), 2022
- [Sombatsiri2019] Sombatsiri, S. et al. "A Design Space Exploration Method of SoC Architecture for CNN-based AI Platform", Synthesis And System Integration of Mixed Information technologies (SASIMI), 2019

Work	HW target	Evaluation speed	Accuracy Timing	Accuracy Power/Energy	Shared resource contention	Inter-layer parallelism	Intra-layer parallelism	Power management	HW dimensions
[Galanis2020]	GPU	✗	✓✓	✓✓	✓	✓	≈	✗	✗
[Tsimpourlas2018]	VPU	✗	✓✓	✓✓	✓	✓	✓	✗	✗
[VelascoMontero2020]	Multicore	✓✓	✓	⊘	✗	✓	≈	✗	✗
[Guo2023]	All	✓✓	✓	✓	✗	✓	≈	✗	✗
[Osterwind2022]	VPU	✓✓	✗	⊘	✗	✓	✓	✗	✗
[Venieris2019]	FPGA	✓✓	✓	⊘	✗	✓	✓	✗	✗
[Parashar2019]	FPGA, GPU	✓✓	✓	✓	✗	✓	✓	✗	✓
[Garbay2023]	MCU	✓✓	✓	✓	✗	✓	✗	✓	✗
[Lee2022]	NPUs	?	✓	⊘	≈	✓	✓	✗	✓
[Sombatsiri2019]	SoC	✗ (~100s)	✓	⊘	✓	✓	✓	✗	✓
THIS WORK	Multicore	✓	✓	✓	✓	✓	✓	✓	✓

Evaluation speed:

- ✓✓ : ~1ms
- ✓ : <60s
- ✗ : >60s

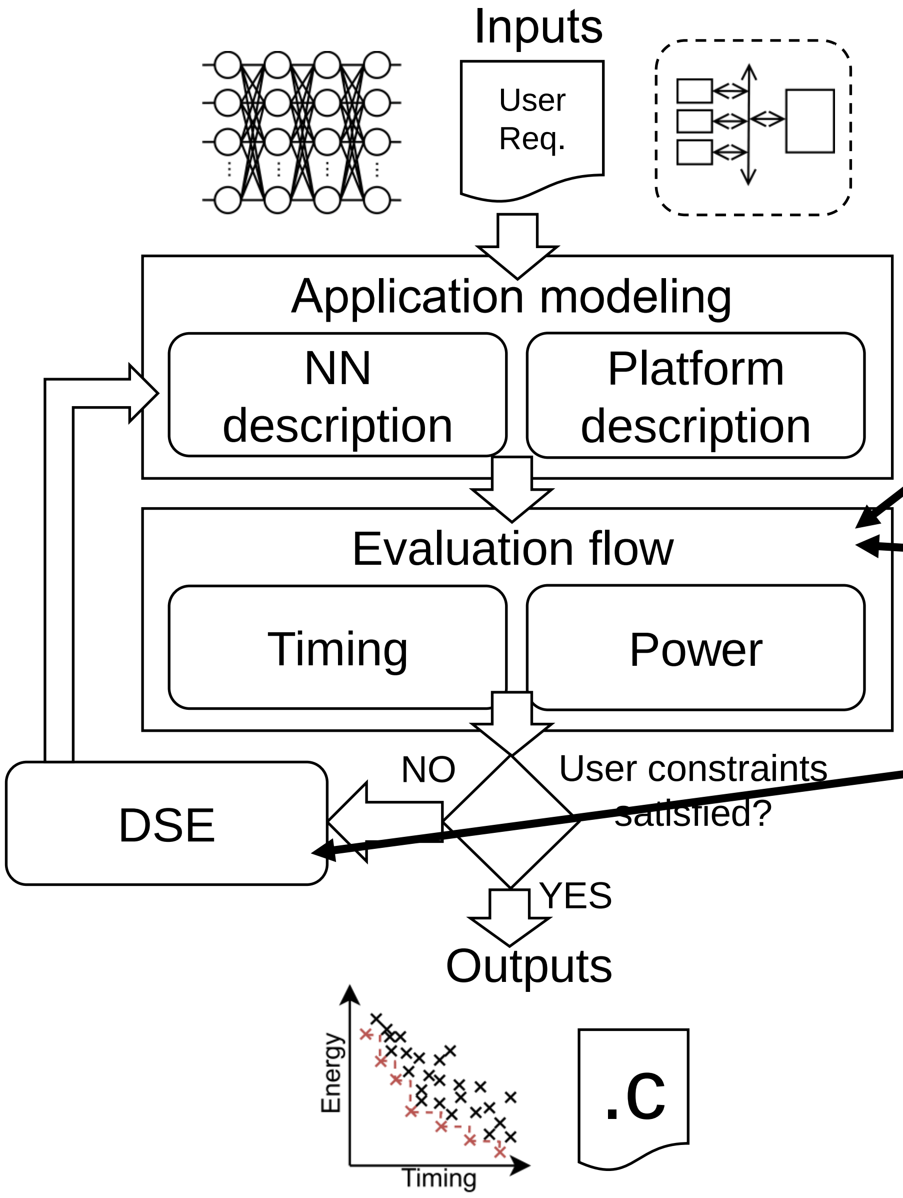
Accuracy:

- ✓✓ : ~100%
- ✓ : >90%
- ✗ : <90%
- ⊘ : N.C.

Other criterias:

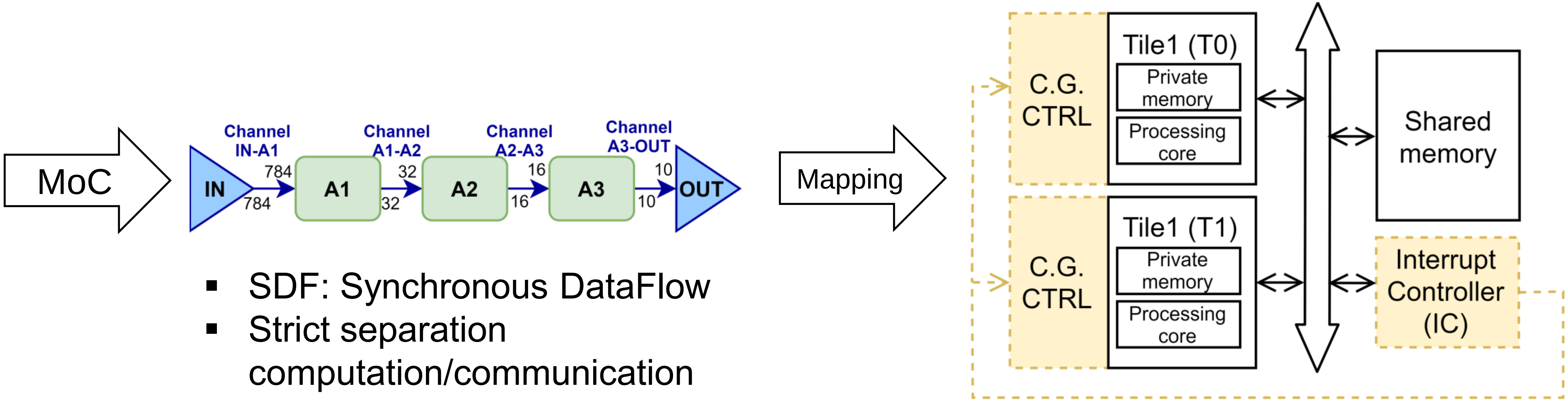
- ✓ : Yes
- ✗ : No
- ≈ : Partial

I. Fundamentals & hypothesis – Research challenges to address



- Research challenges:
  1. How to provide fast yet accurate evaluation early in design phases of timing and energy properties for streaming NNs deployments on multi-core platforms?
  2. Is a model-based approach more relevant than rapid prototyping?
  3. Is a model-based approach suited for early, fast and confident Design Space Exploration (DSE) of streaming NNs deployments on multi-core platforms?

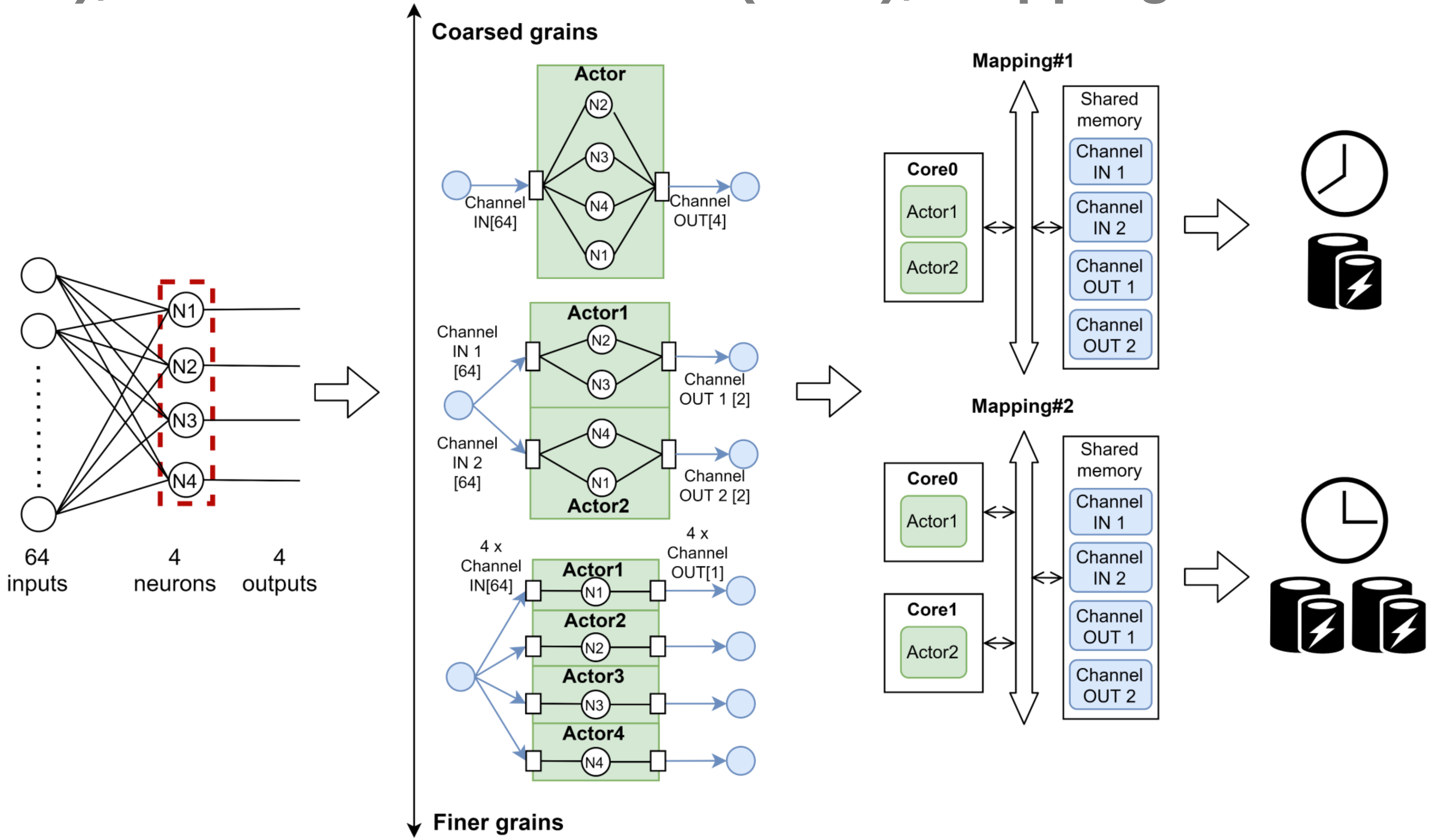
I. Fundamentals & hypothesis – Model of Computation (MoC), Model of Architecture (MoA), mapping



- SDF: Synchronous DataFlow
- Strict separation computation/communication
  - Actors,
  - Channels,
  - Tokens.

- MoA: Model of Architecture
- Two versions:
  - Without power management: polling
  - With power management: interrupt + clock gating

I. Fundamentals & hypothesis – Model of Computation (MoC), Model of Architecture (MoA), mapping

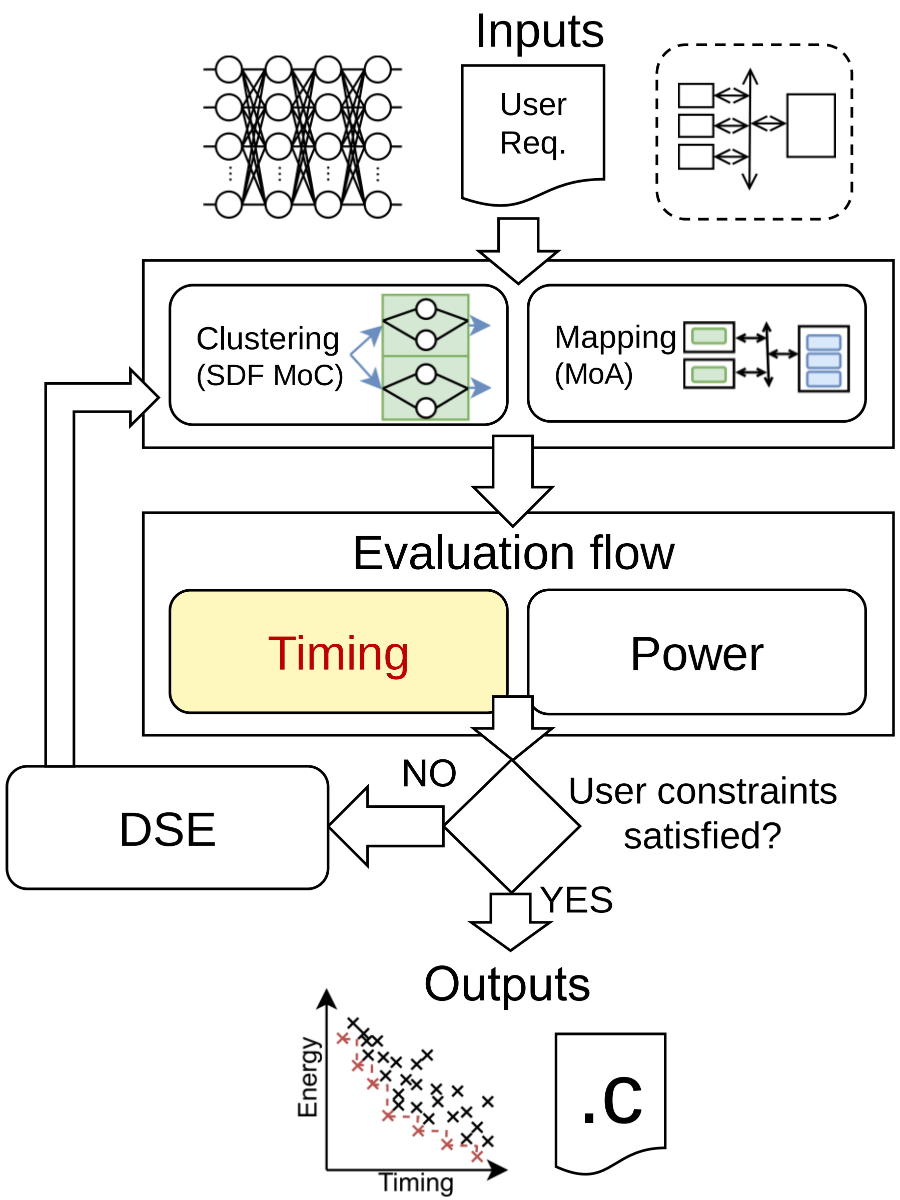


Artificial Neural Network (NN)

Clustering (description using a dataflow-oriented MoC)

Mapping

Performance/power analysis



I. Related work & work hypothesis

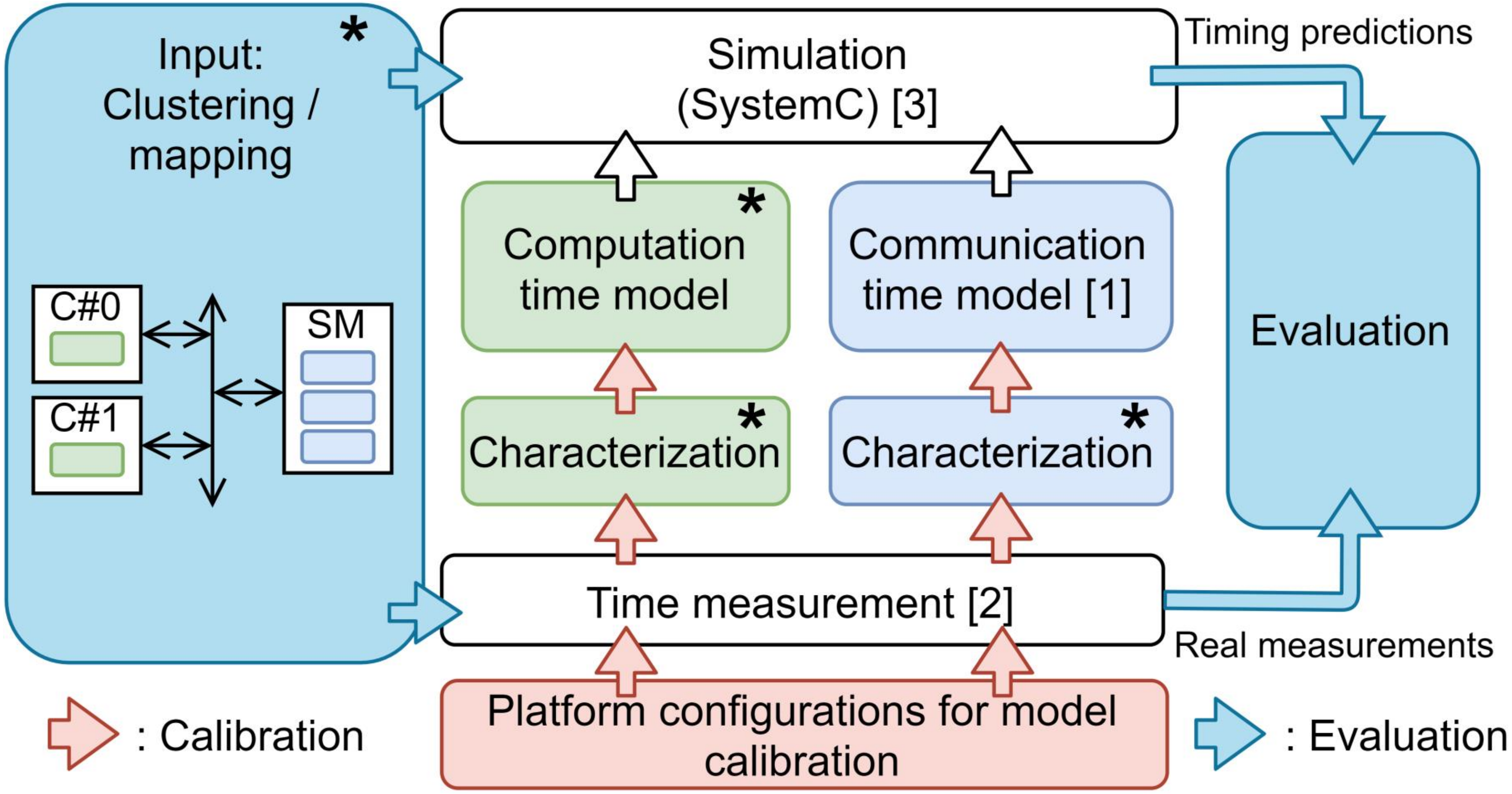
II. Timing prediction flow

III. Power and energy analysis flow

IV. Design Space Exploration flow

V. Conclusion & Prospects

II. Timing modeling flow - Overview

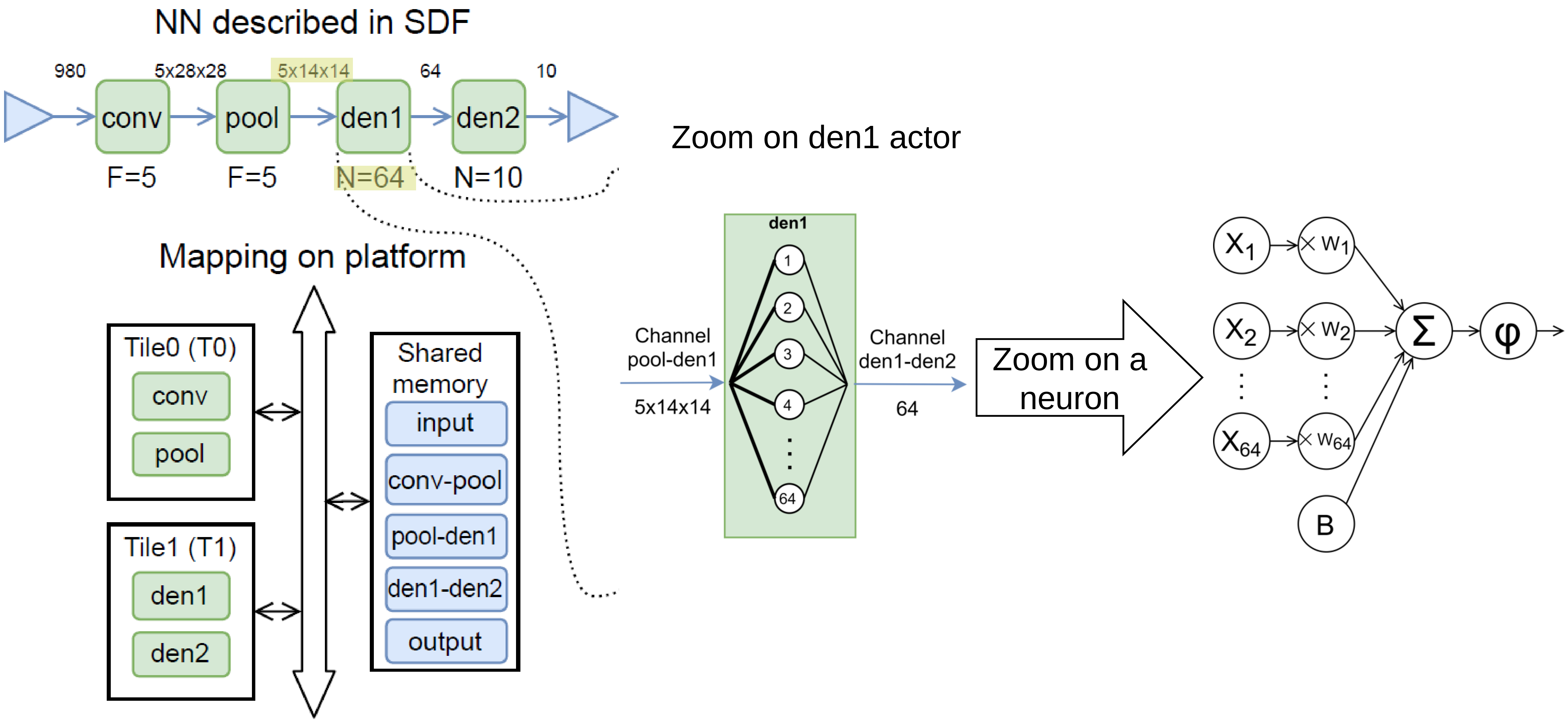


[1] Vu, H.-D. "Fast and Accurate Performance Models for Probabilistic Timing Analysis of SDFGs on MPSoCs", PhD thesis, *Université de Nantes,* 2021

[2] Schlaak, C.; Fakih, M. & Stemmer, R. “Power and Execution Time Measurement Methodology for SDF Applications on FPGA-based MPSoCs”, *International Workshop on High Performance Energy Efficient Embedded Systems (HIP3ES),* 2017

[3] Stemmer, R.; Vu, H.-D.; Le Nours, S.; Grüttner, K.; Pillement, S. & Nebel, W. “A Measurement-Based Message-Level Timing Prediction Approach for Data-Dependent SDFGs on Tile-Based Heterogeneous MPSoCs”, *Applied Sciences,* 2021

II. Timing modeling flow – Computation time model

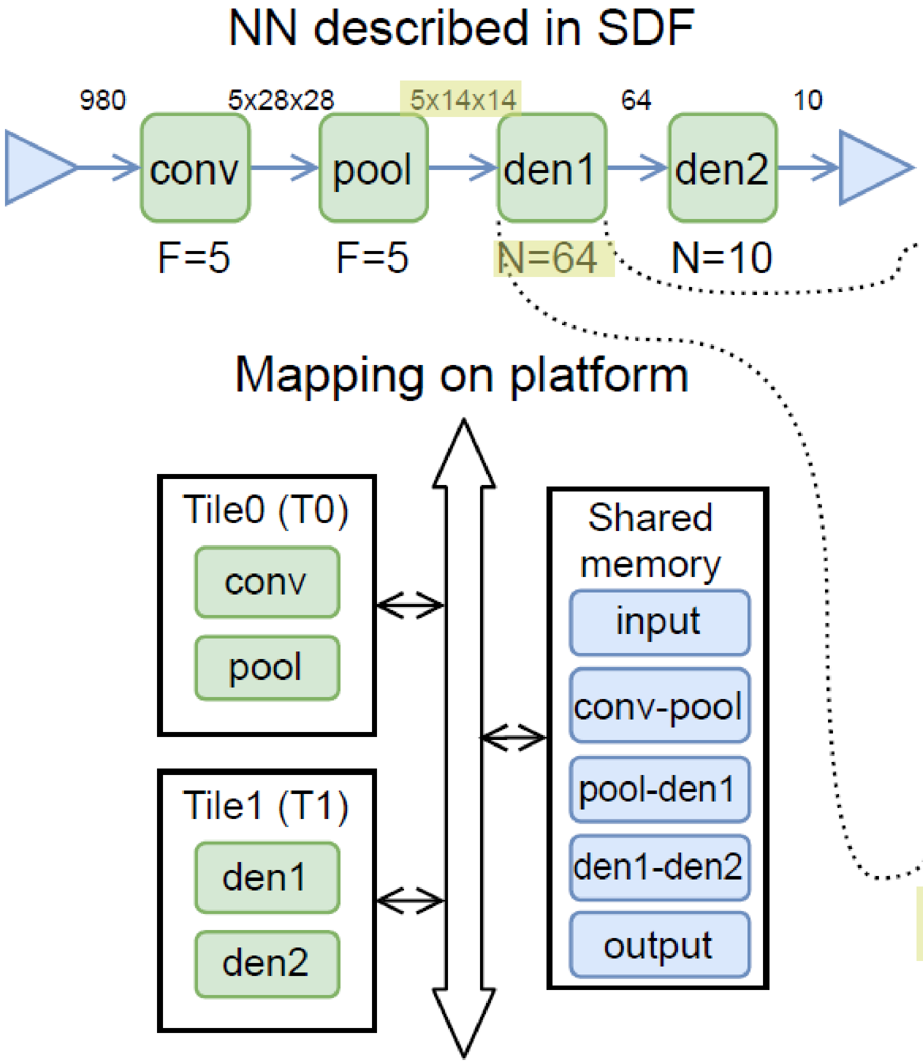


NN described in SDF

Mapping on platform

Zoom on den1 actor

II. Timing modeling flow – Computation time model



den1 actor pseudo-code

```
WHILE(1):
    ...
  ReadTokens(pool-den1);
  ExecuteActorDen1(N=5x14x14, M=64, FLOAT
                   input[N],FLOAT output[M]):
    INTEGER m;
    FOR m FROM 0 TO M-1:
        INTEGER n; FLOAT sum = 0;
        FOR n FROM 0 TO N-1:
          sum <= sum + weights[m][n]*input[n];
        sum <= sum + bias;
        output[m] <= ActivationFunction(sum);
  WriteTokens(den1-den2);
    ...
```

D_{setup}

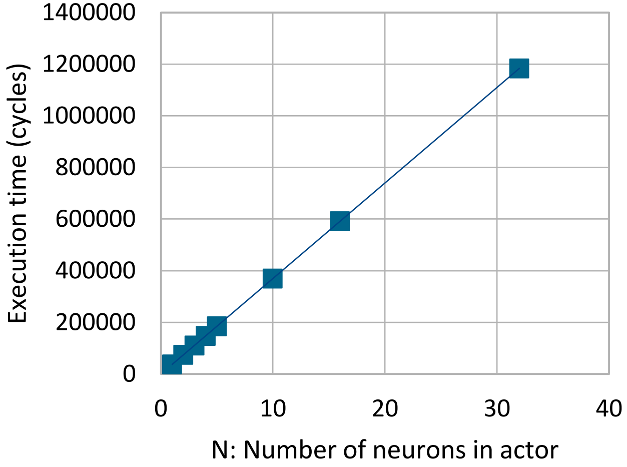
D_{Σ} D_{φ}

N: number of neurons in cluster, M: number of inputs of layer

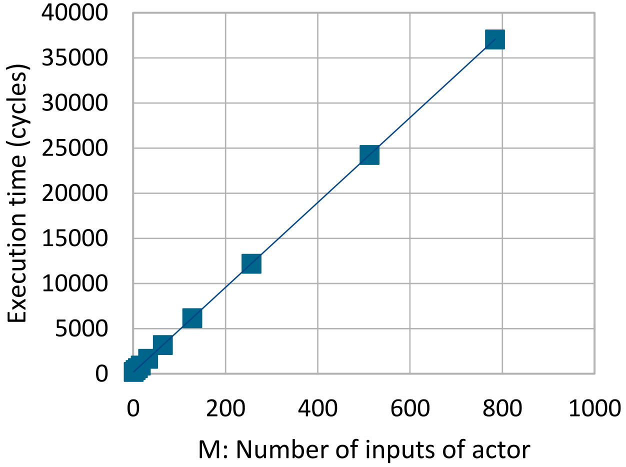
$$D_{dense}(M,N) = M \cdot N \cdot D_{\Sigma} + M \cdot D_{\varphi} + D_{setup}$$

II. Timing modeling flow – Model calibration

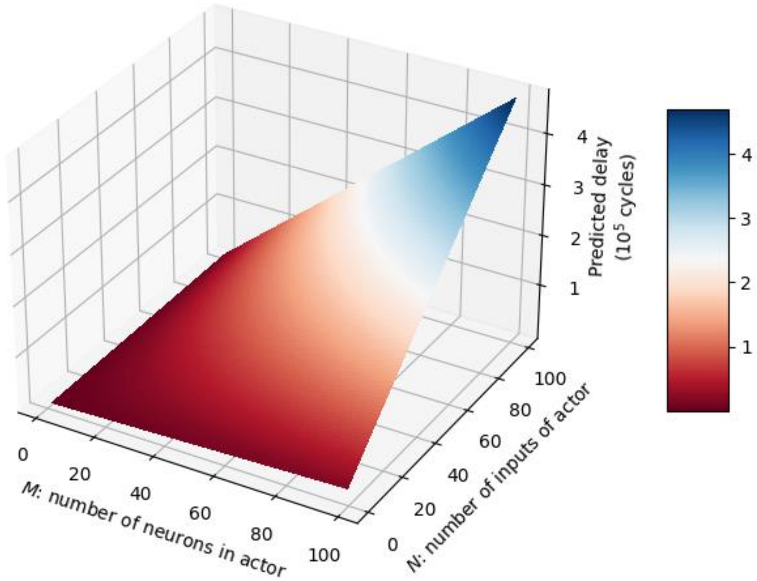
Evolution of cluster execution time while varying N (fixed M=784)



Evolution of cluster execution time while varying M (fixed N=1)



Predicted actor computation time in cycles based on M and N



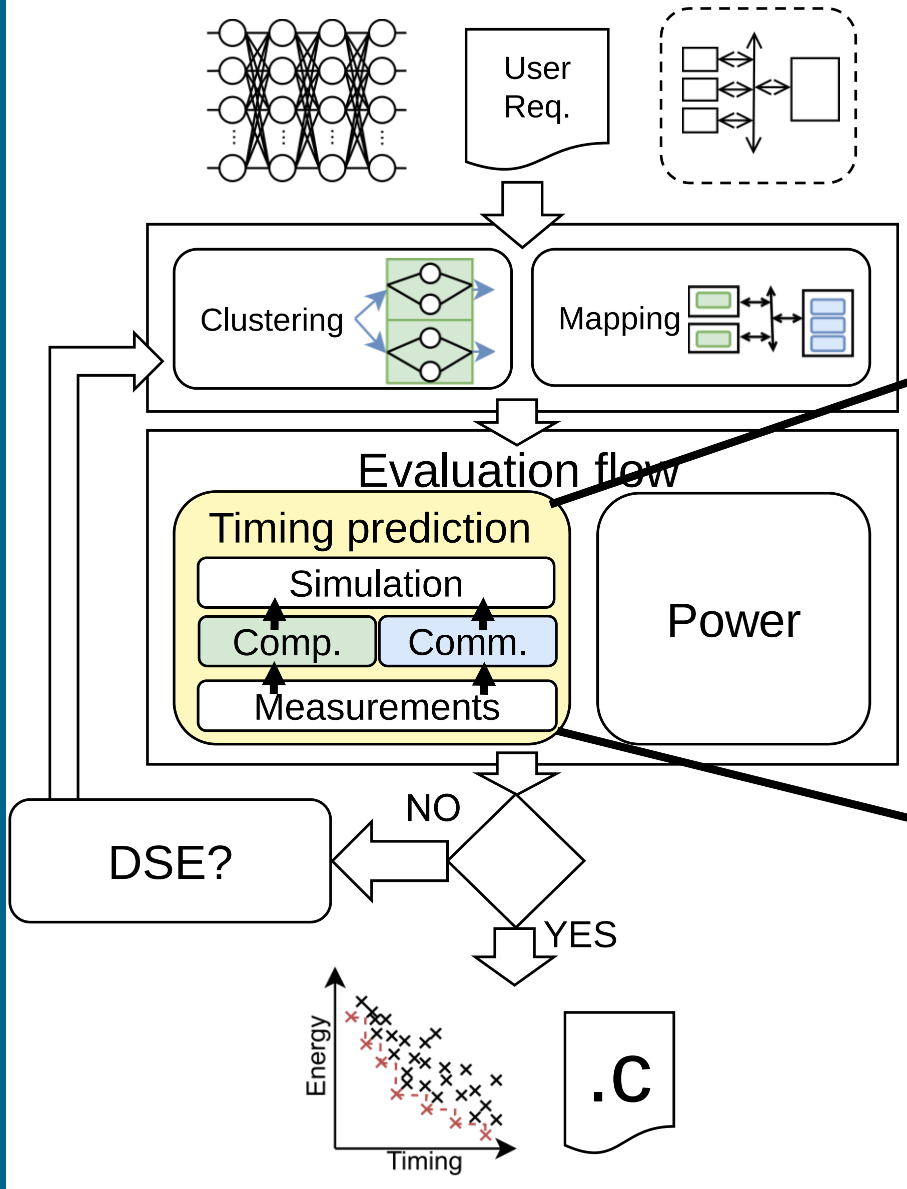
$$D_{dense}(M,N) = M \cdot N \cdot D_{\Sigma} + M \cdot D_{\varphi} + D_{setup}$$

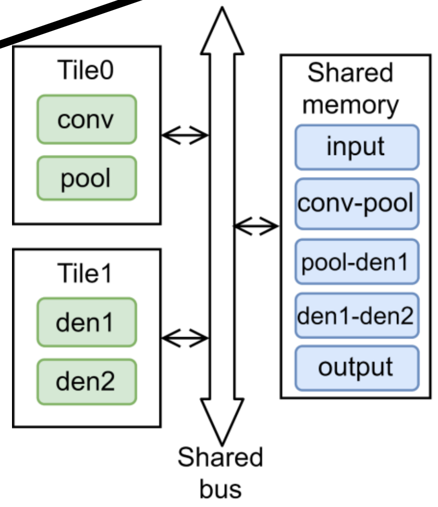
$$D_{\Sigma} = 47^{*},\ D_{\varphi} = 146^{*} \text{and } D_{setup} = 39^{*}$$

Estimated: $D_{\Sigma} = 30^{*},\ D_{\varphi} = 61^{*} \text{and } D_{setup} = 6^{\ *}$

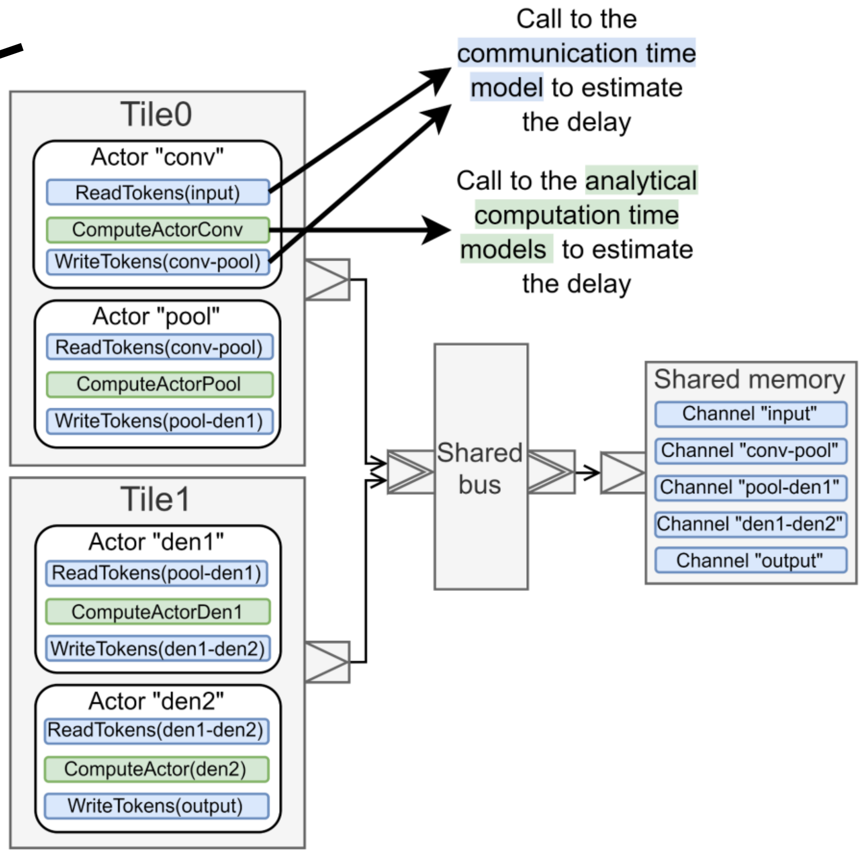
*: value in processor cycles

III. Timing modeling flow – Simulation described in SystemC





Mapping of NN

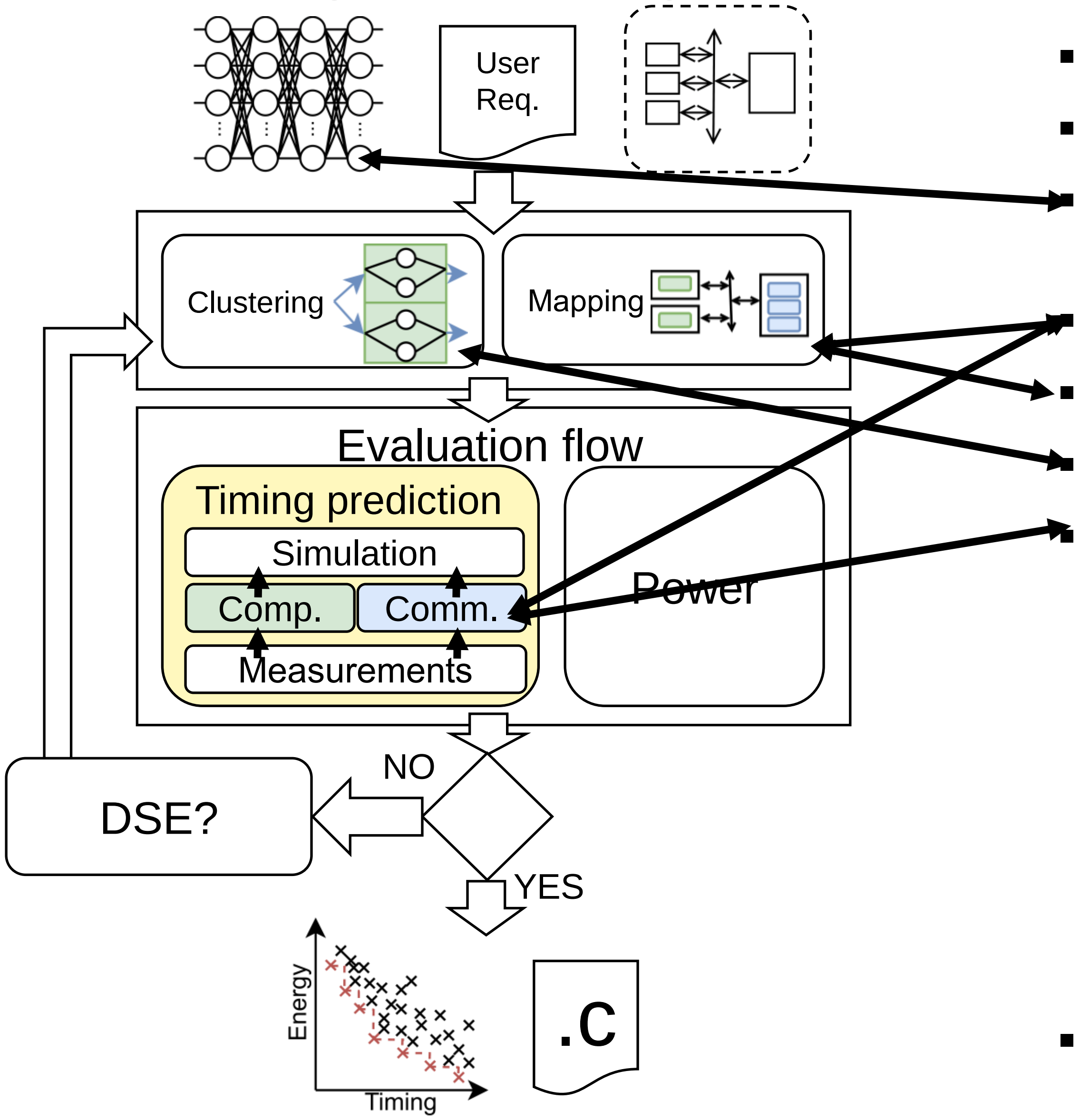


Call to the communication time model to estimate the delay

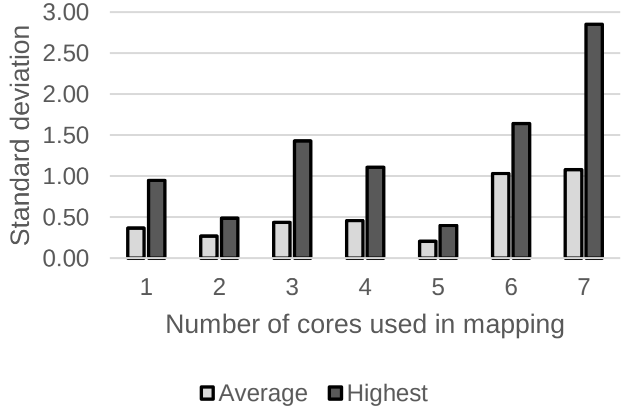
Call to the analytical computation time models to estimate the delay

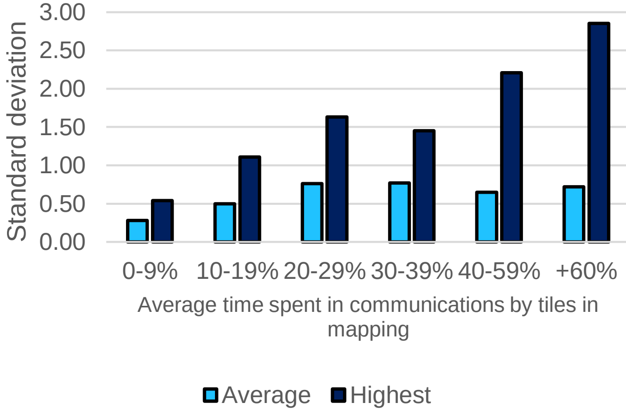
Setup of the SystemC simulation for the considered mapping

III. Timing modeling flow – Experimental results, test cases

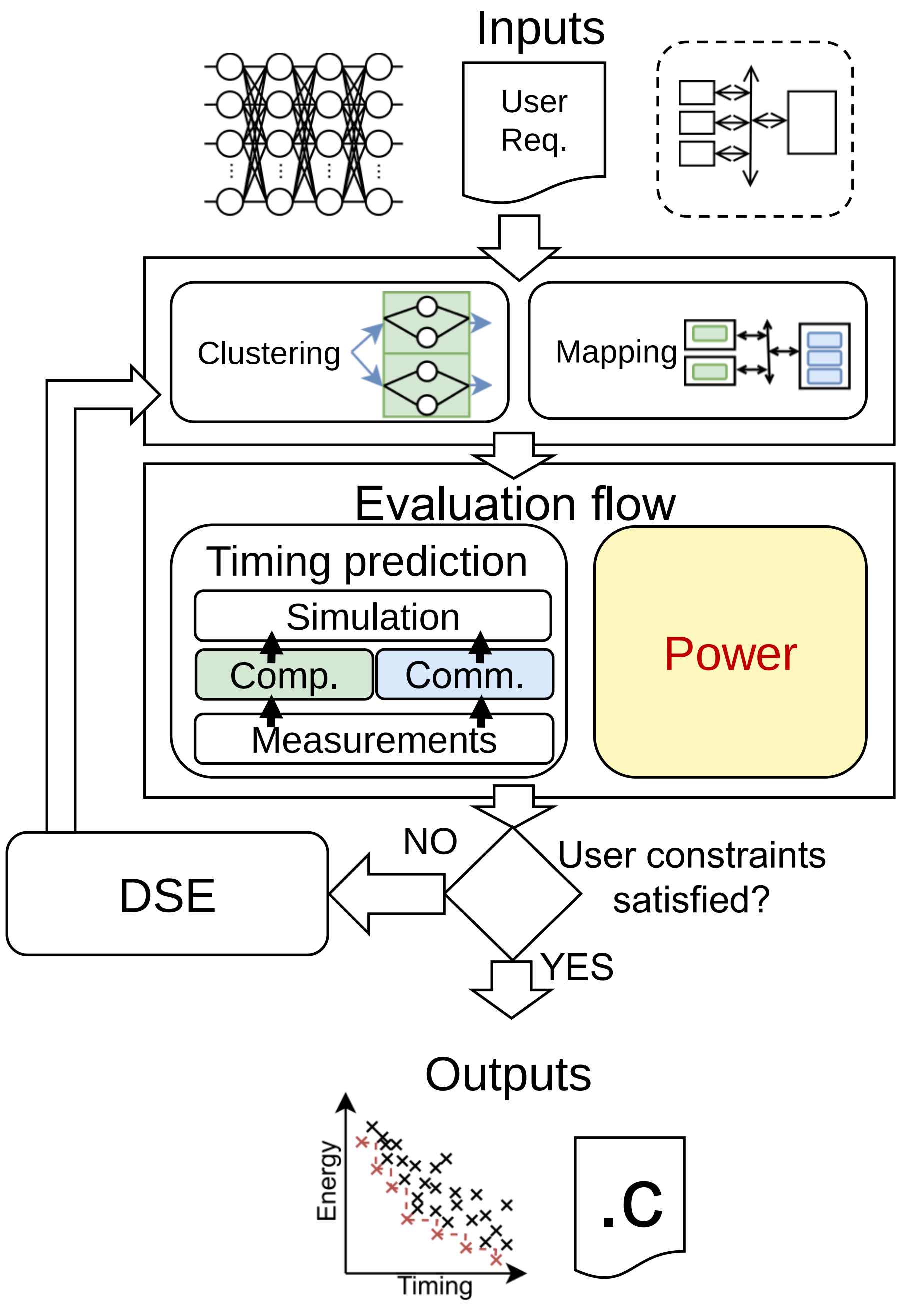


- 1 - Overall accuracy: >97% on 54 mappings.
- 2 - Evaluation speed: ~20s.
- 3 - NN different workloads ✓
  - MLP1: 0,83%, MLP2: 0,31%, MLP3: 0,62%, CNN1: 0,43%. OK
- 4 - Communication procedure (polling or interrupt) ✓
- 5 - Number of cores used: ✓
- 6 - NN clustering complexity: ✓
- 7 - Communication rates: ✓





- 8 - Comparison with analytical model:
  - Error up to 30% on multi-core scenarios.
  - Very high evaluation speed (~1ms)



I. Related work & work hypothesis

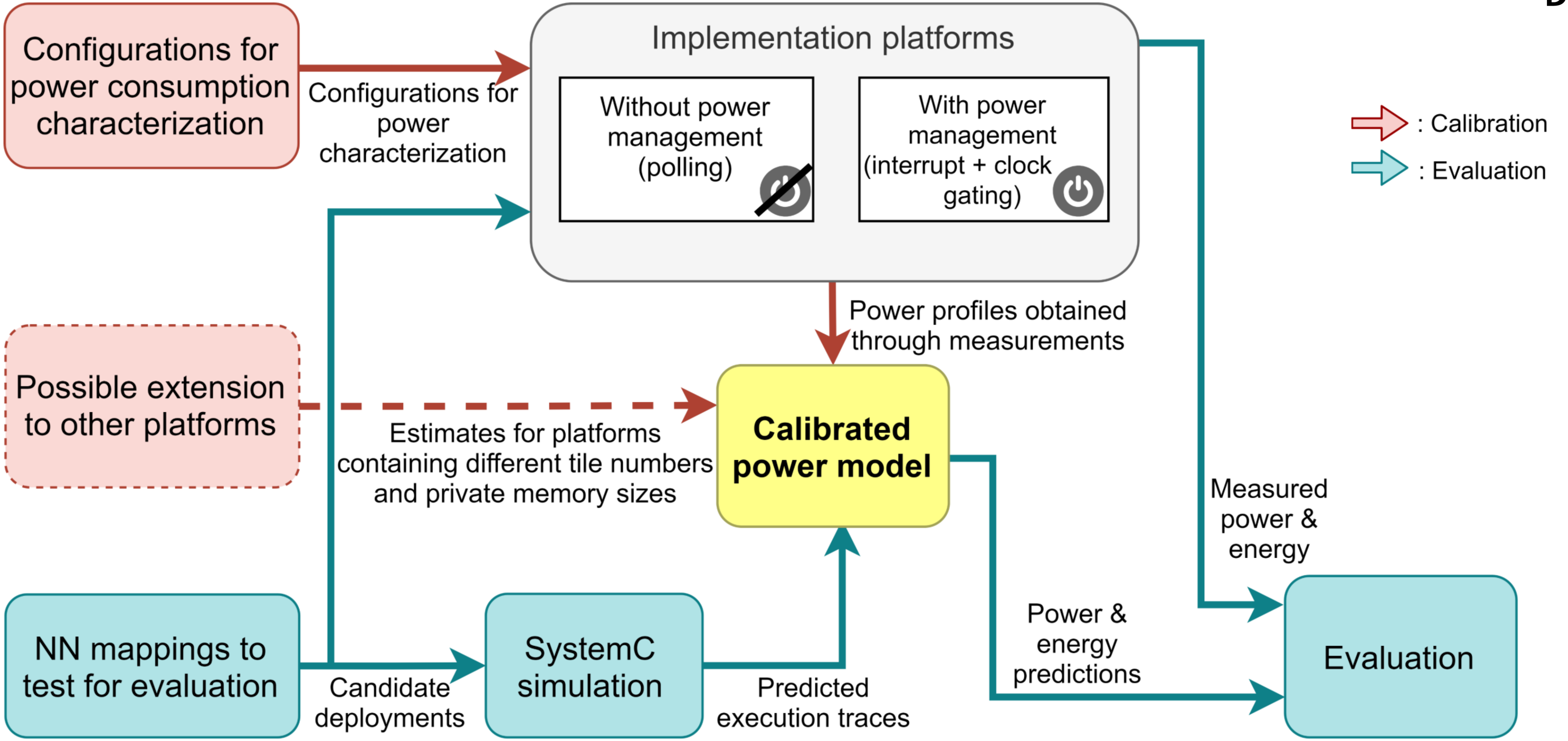
II. Timing prediction flow

III. Power and energy analysis flow

IV. Design Space Exploration flow

V. Conclusion & Prospects

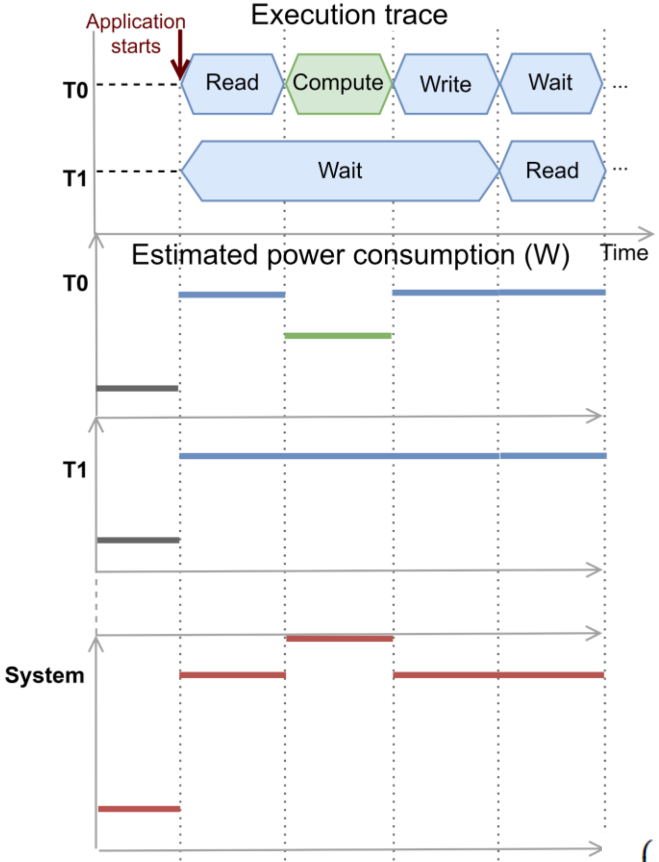
IV. Power modeling flow - Overview


Configurations for power consumption characterization
Configurations for power characterization
Implementation platforms
Without power management (polling)
With power management (interrupt + clock gating)
Power profiles obtained through measurements
Possible extension to other platforms
Estimates for platforms containing different tile numbers and private memory sizes
Calibrated power model
NN mappings to test for evaluation
Candidate deployments
SystemC simulation
Predicted execution traces
Power & energy predictions
Measured power & energy
Evaluation
: Calibration
: Evaluation

IV. Power modeling flow – Proposed model

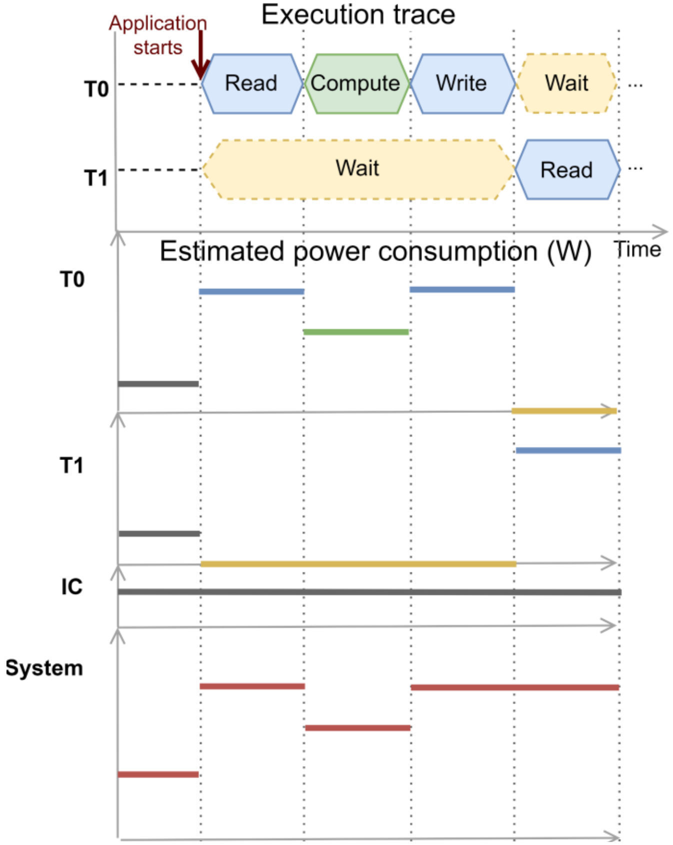
$$P(t) = \underline{P_{\text{static}}} + \underline{P_{\text{comp}}(t)} + \underline{\underline{P_{\text{comm}}(t)}}$$

Without power management △



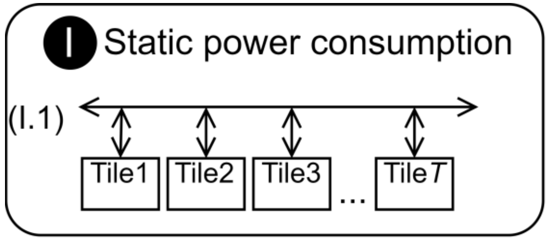
$$P_{\triangle,\text{comm}}(t) = \underline{\underline{P_{\triangle,\text{rwp}}(t)}} = \begin{cases} P_{\text{sm}} & \text{if at least one tile is reading, writing} \\ & \text{or polling on shared memory at time } t \\ 0 & \text{otherwise} \end{cases}$$

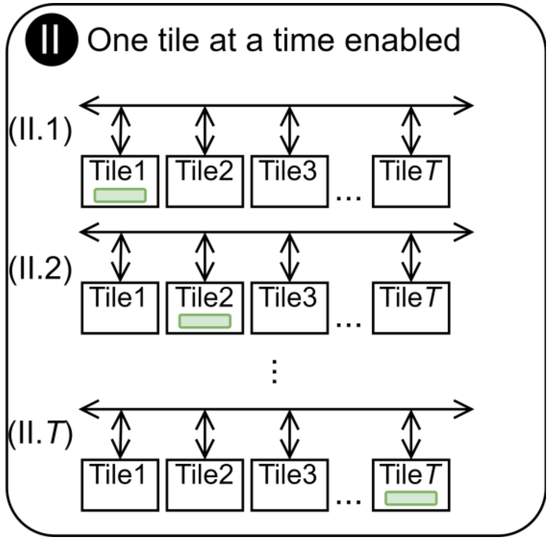
With power management ▲

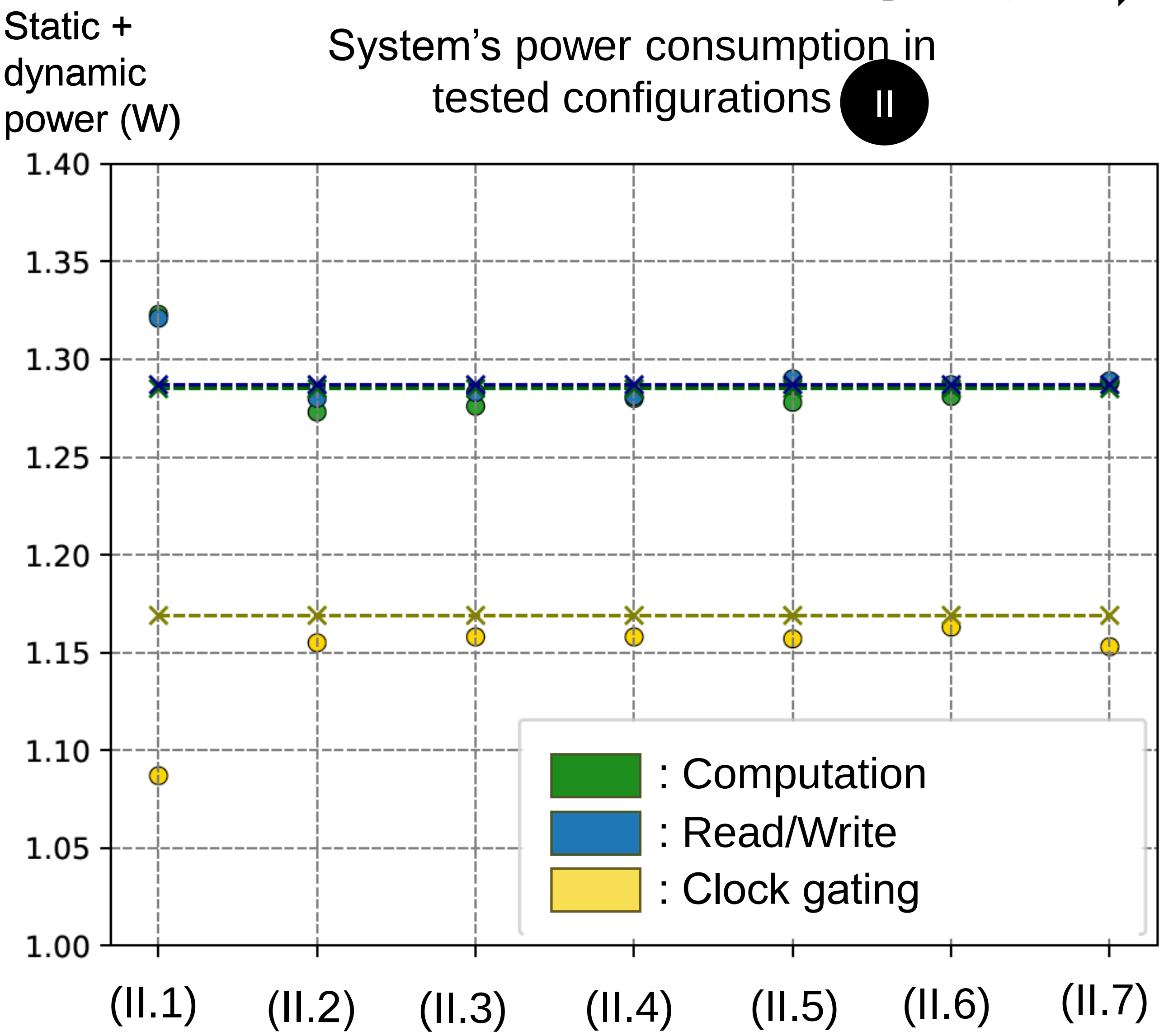


$$P_{\blacktriangle,\text{comm}}(t) = \underline{P_{\blacktriangle,\text{rw}}(t)} + \underline{P_{\blacktriangle,\text{cg}}(t)}$$

IV. Power modeling flow – Power model calibration



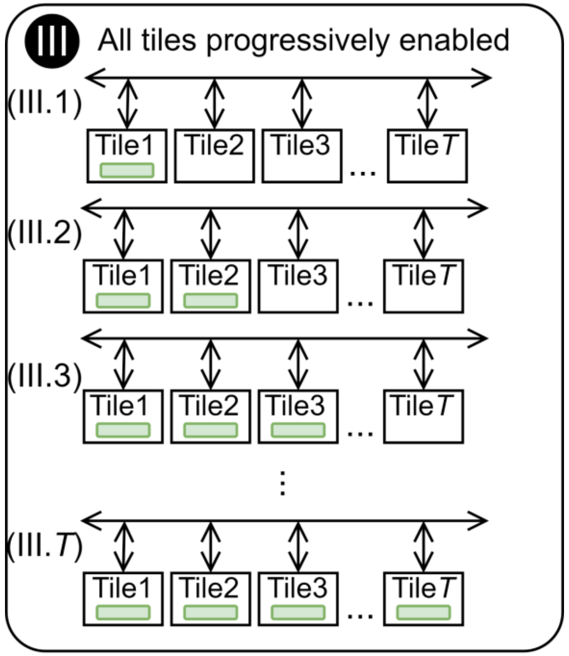


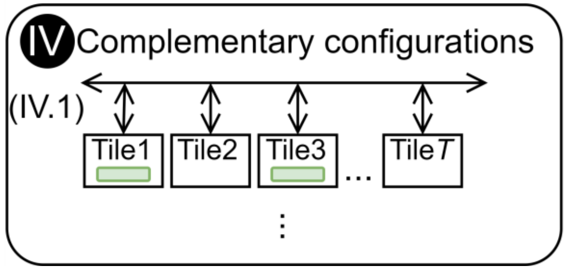


IV. Power modeling flow – Power model calibration

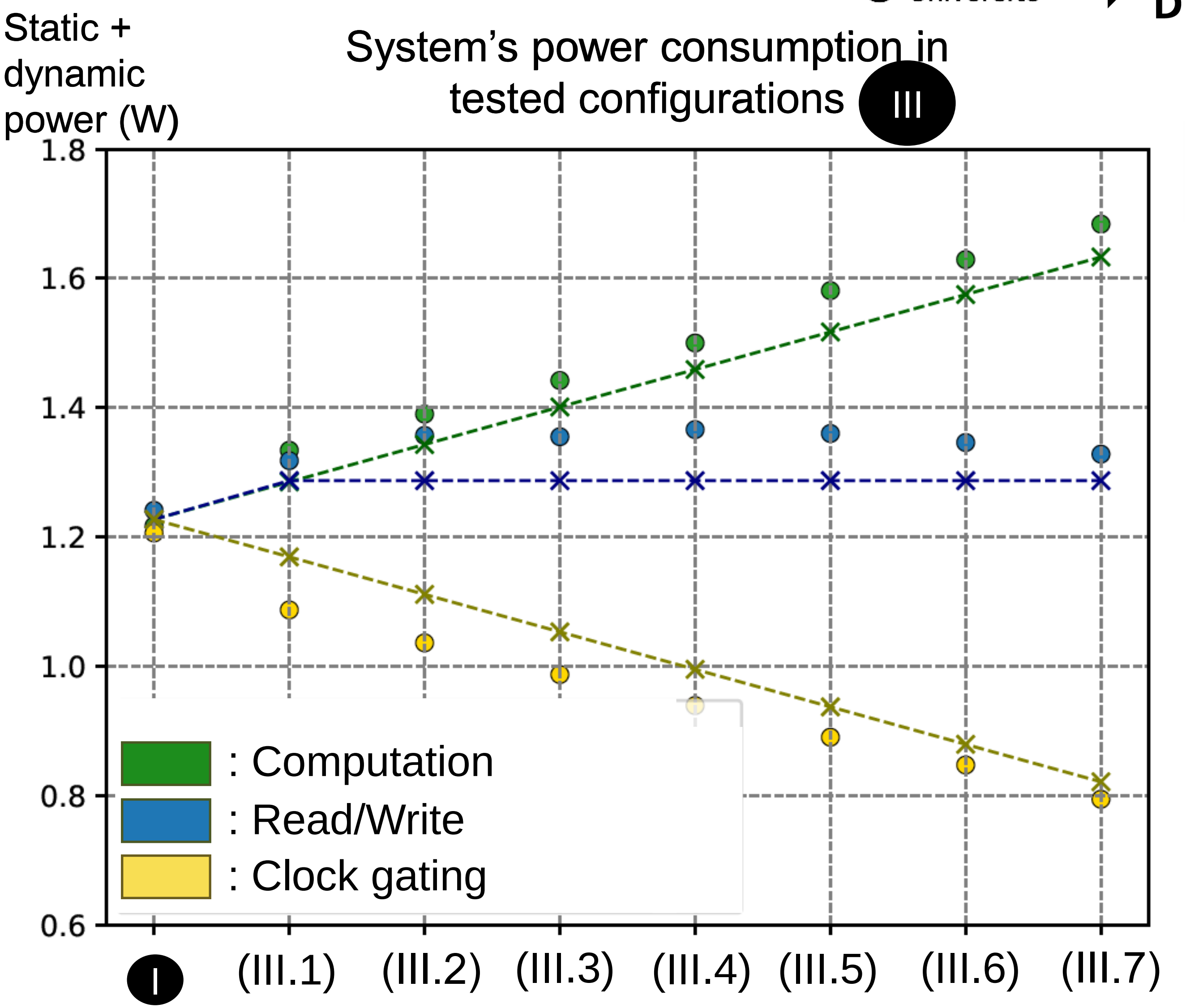
I Static power consumption

II One tile at a time enabled

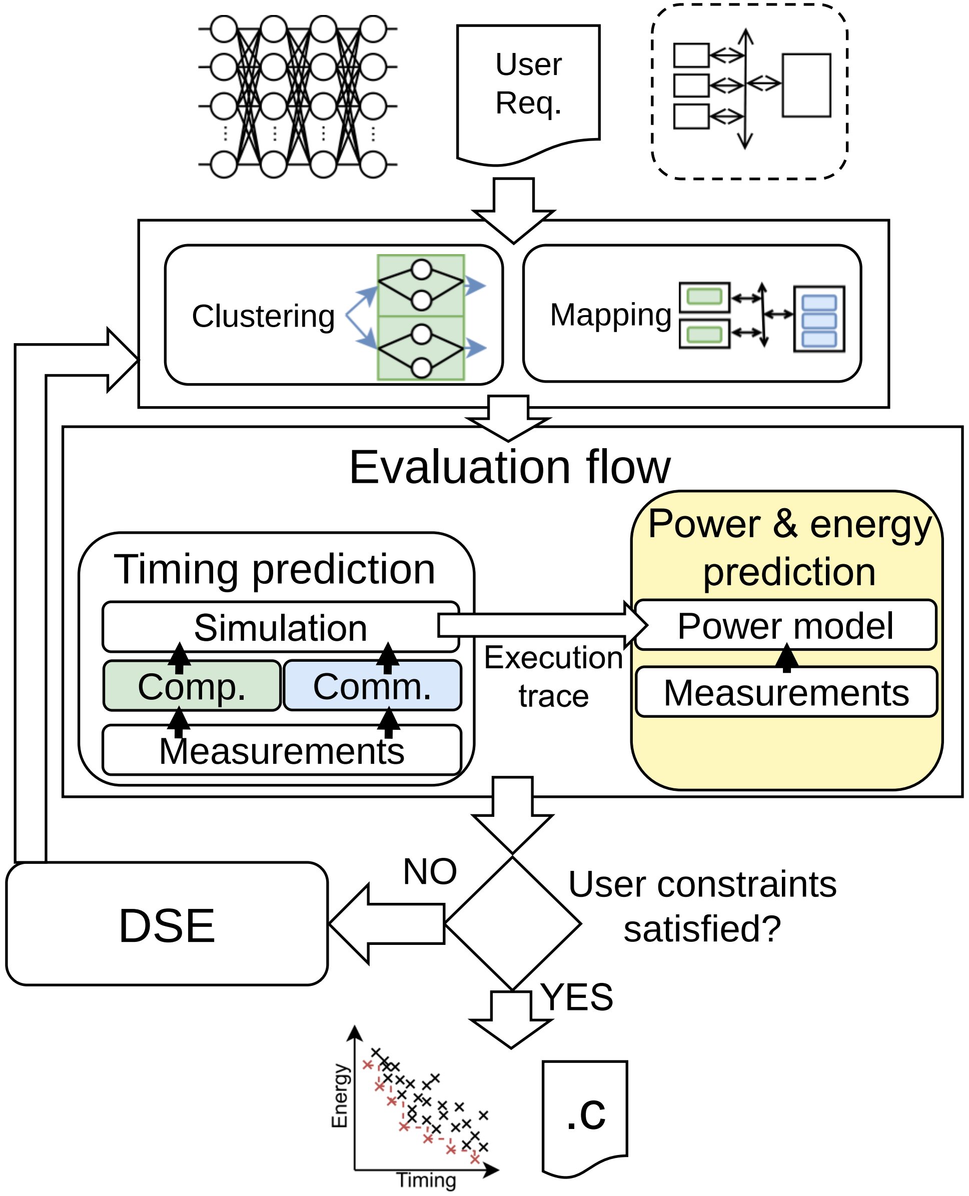




System's power consumption in tested configurations III



IV. Power modeling flow – Experiments



User Req.

Clustering

Mapping

Evaluation flow

Timing prediction

Simulation

Comp.

Comm.

Measurements

Execution trace

Power & energy prediction

Power model

Measurements

User constraints satisfied?

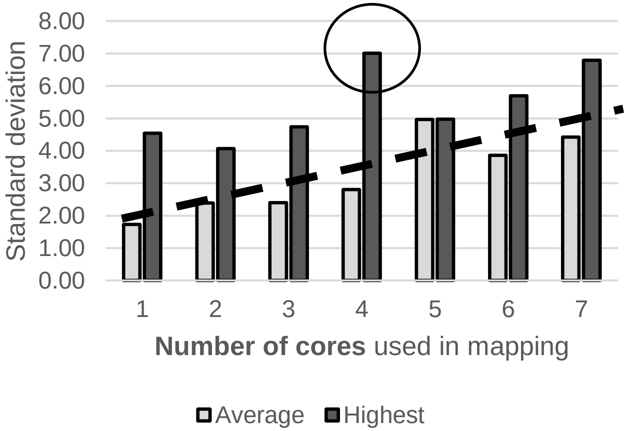
NO

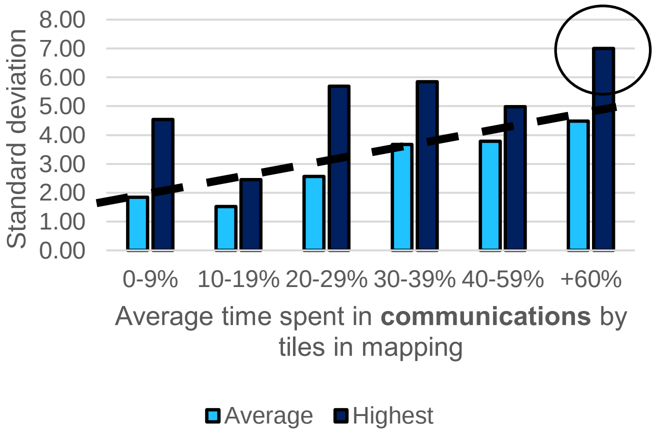
YES

DSE

.c

- 1 - Overall accuracy: >93% on 54 mappings.
- 2 - Evaluation speed: ~20s.
- 3 - NN different workloads: average prediction error between 1,8% and 3% for the 4 NNs ✓
- 4 - Use of power management: average is 2,11% without, 3,92% with ✓
- 5 - Number of cores used and communication rates ✓
- 6 - Analytical model:
  - Maximum error: ~20%
  - Evaluation time: ~1ms

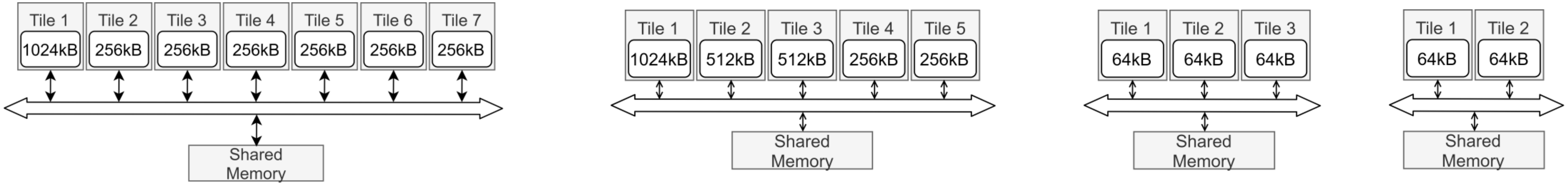




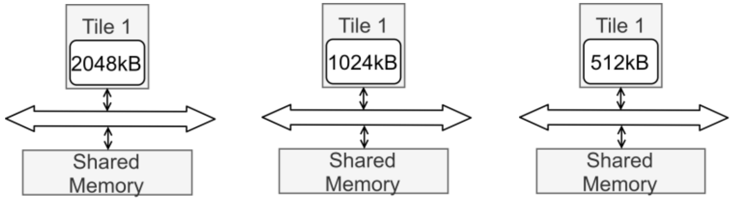
IV. Power modeling flow – Experiments

- Use to jointly evaluate and optimize multi-core platform architectures and NN deployments under power and energy constraints

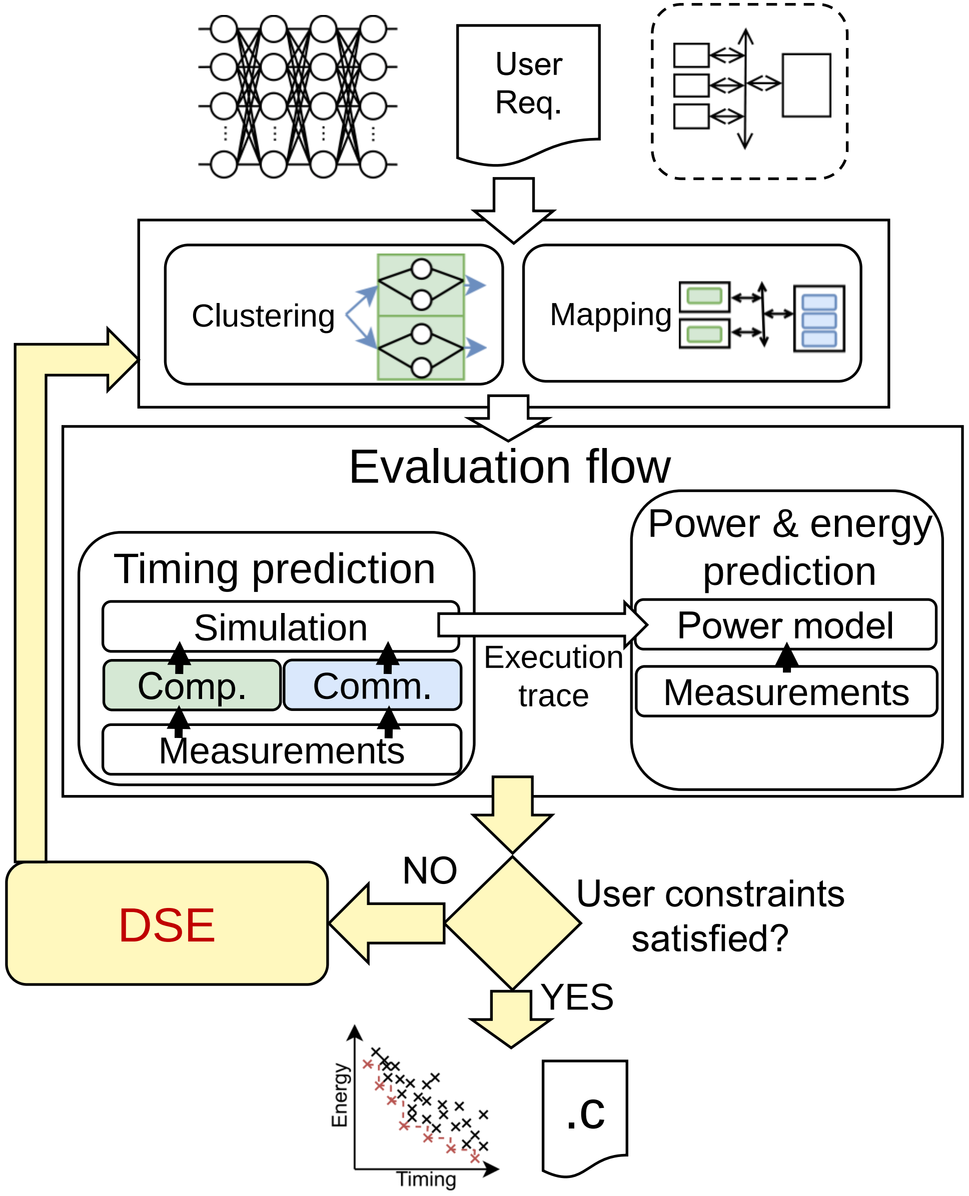
Multi-core platform versions:



Single-core platform versions:



Static power consumption only		Static + dynamic	
Multi-core	Single-core	Multi-core	Single-core
< 5%	< 5%	~ 5%	**> 10%**



I. Related work overview

II. Technical background

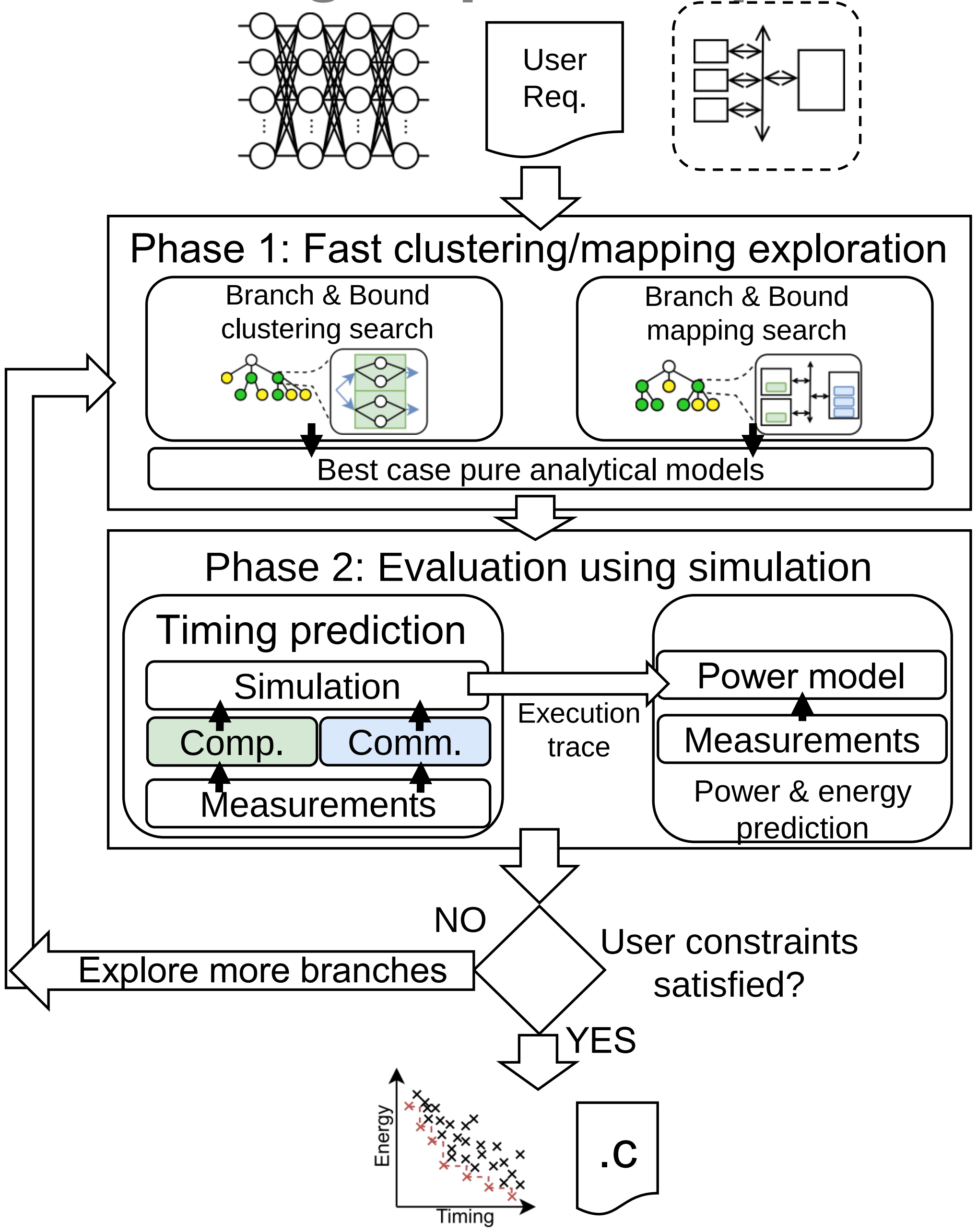
III. Timing prediction flow

IV. Power and energy analysis flow

V. Design Space Exploration flow

VI. Conclusion & Prospects

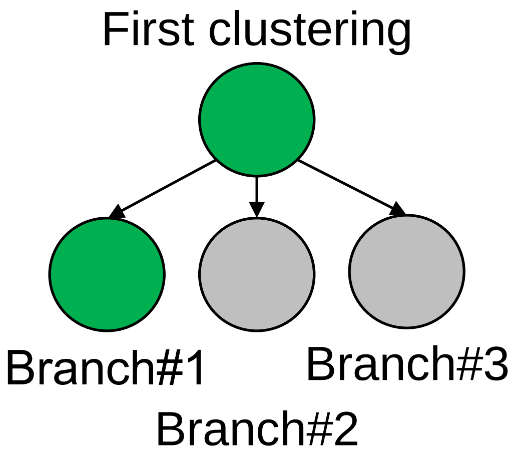
V. Design Space Exploration (DSE) - Overview

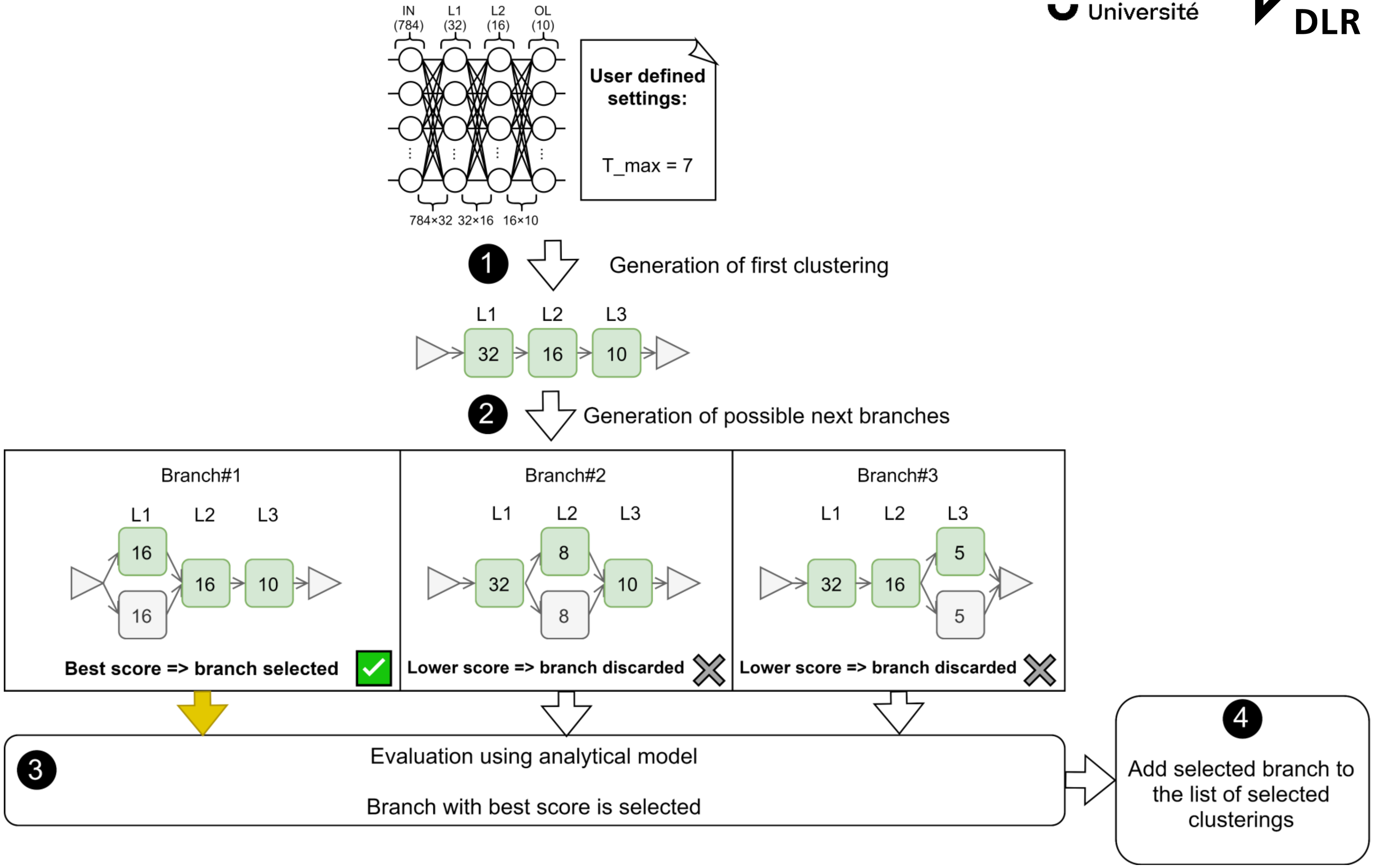


User Req.
Phase 1: Fast clustering/mapping exploration
Branch & Bound clustering search
Branch & Bound mapping search
Best case pure analytical models
Phase 2: Evaluation using simulation
Timing prediction
Simulation
Comp.
Comm.
Measurements
Execution trace
Power model
Measurements
Power & energy prediction
NO
Explore more branches
User constraints satisfied?
YES
Energy
Timing
.C

- 2 phases:
  - Phase 1: Fast exploration using best case pure analytical models
  - Phase 2: Slower but accurate evaluation of most relevant mappings using simulation.
- Branch & Bound enhanced clustering and mapping search
- Possibility to perform several iterations of the flow in order to consider additional branches.

V. DSE – Branch & Bound enhanced **clustering** search





User defined settings:

T_max = 7

1 Generation of first clustering

2 Generation of possible next branches

Branch#1: **Best score => branch selected** ✓

Branch#2: **Lower score => branch discarded** ✗

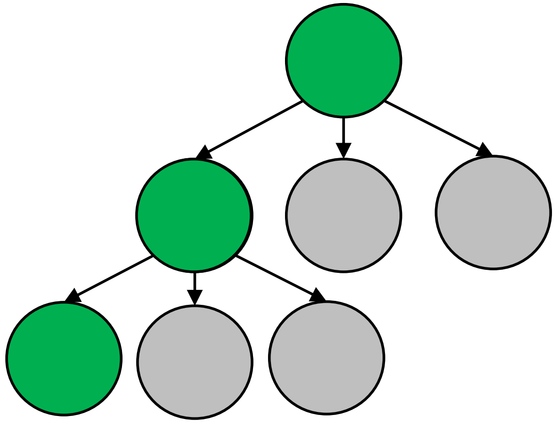
Branch#3: **Lower score => branch discarded** ✗

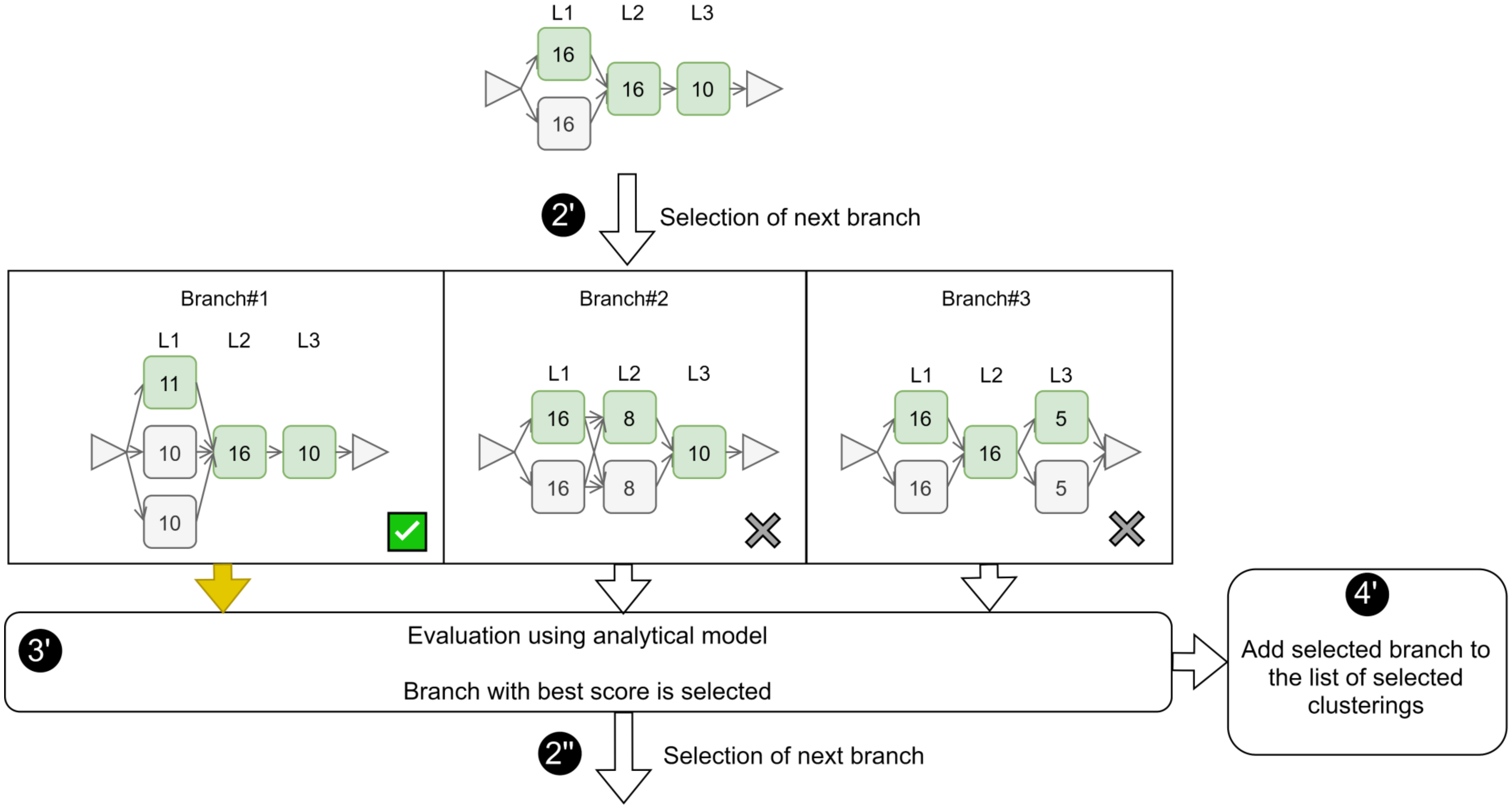
3 Evaluation using analytical model

Branch with best score is selected

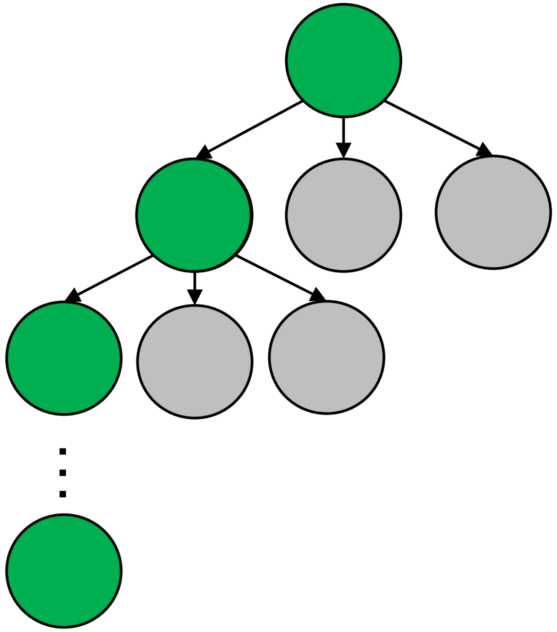
4 Add selected branch to the list of selected clusterings

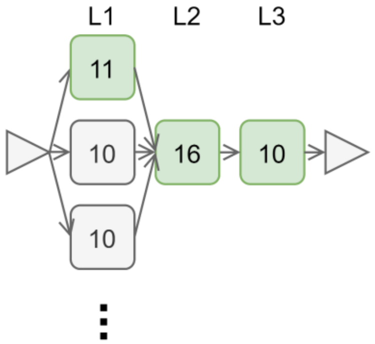
V. DSE – Branch & Bound enhanced clustering search



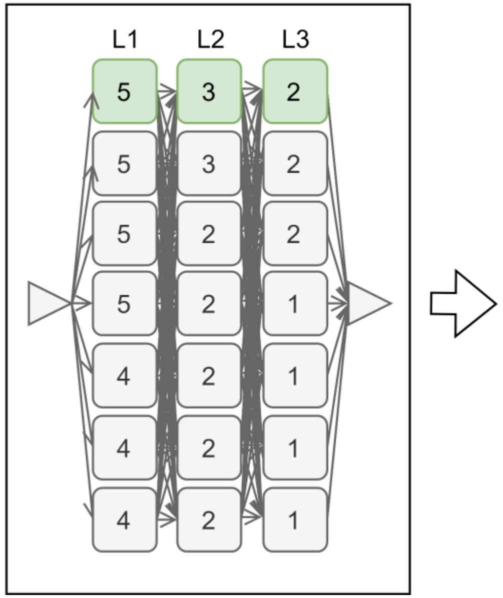


V. DSE – Branch & Bound enhanced **clustering** search





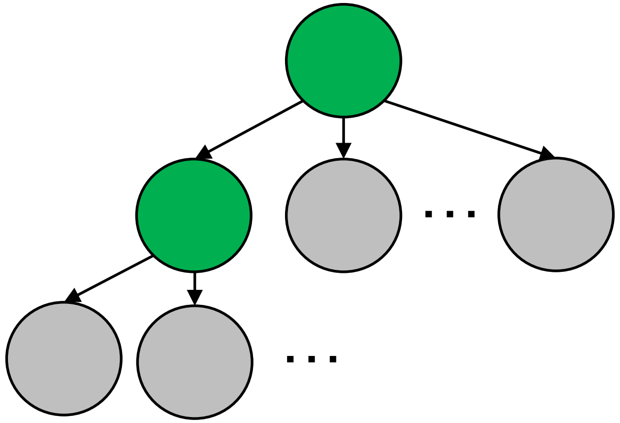
5 Final clustering is reached
(no more evolution possible)

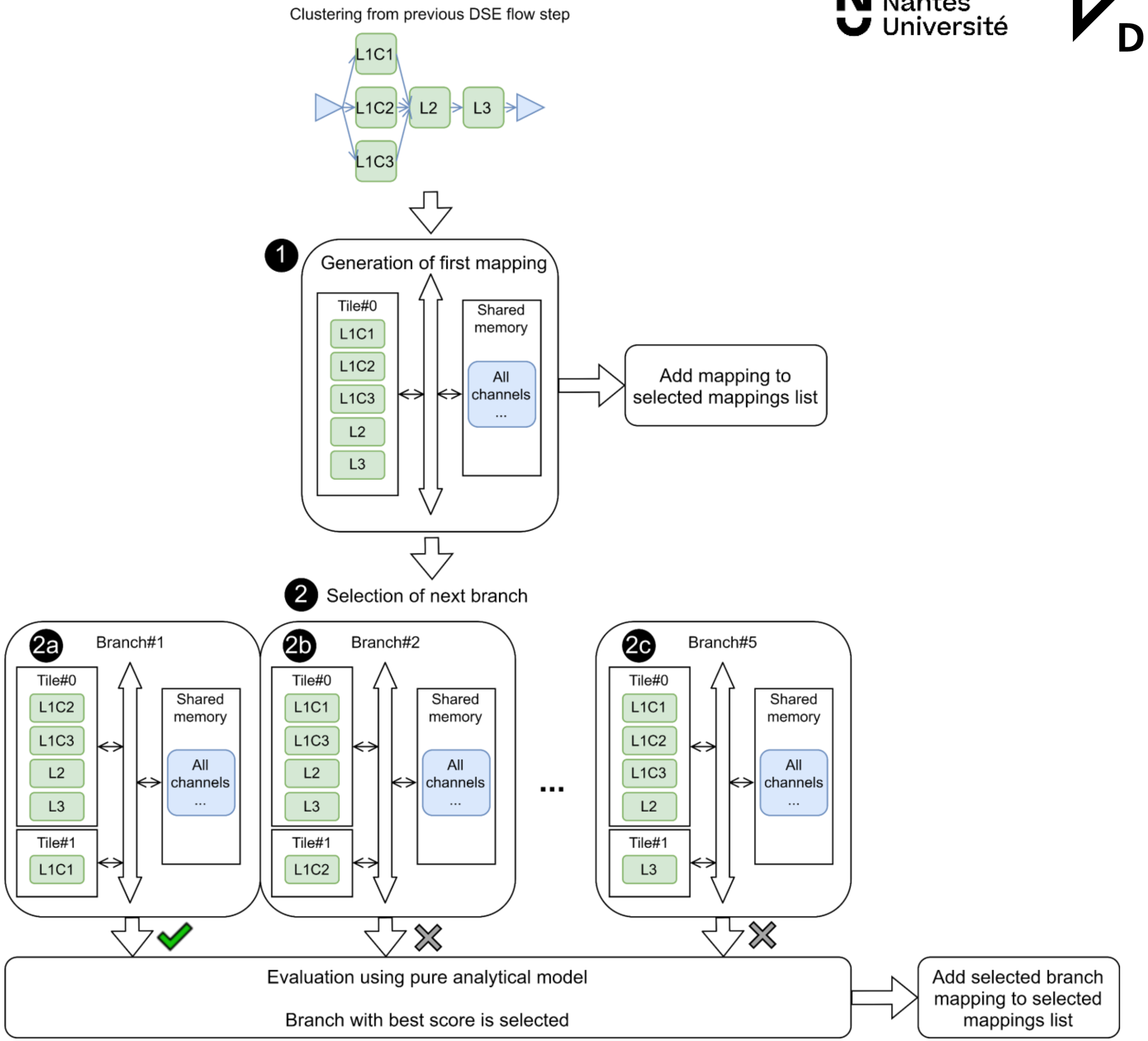


L1	L2	L3
5	3	2
5	3	2
5	2	2
5	2	1
4	2	1
4	2	1
4	2	1

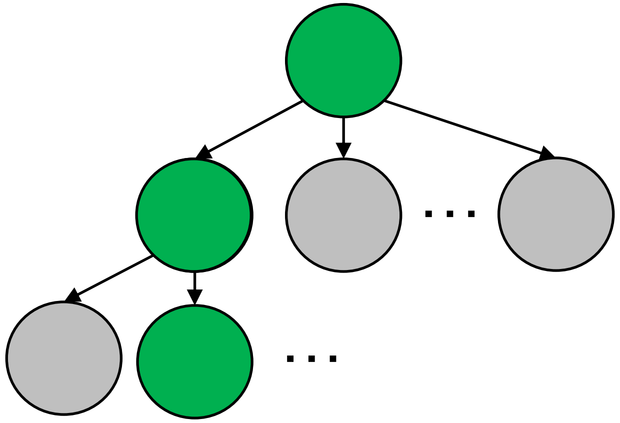
6 Add final clustering to the list of selected clusterings and return the list.

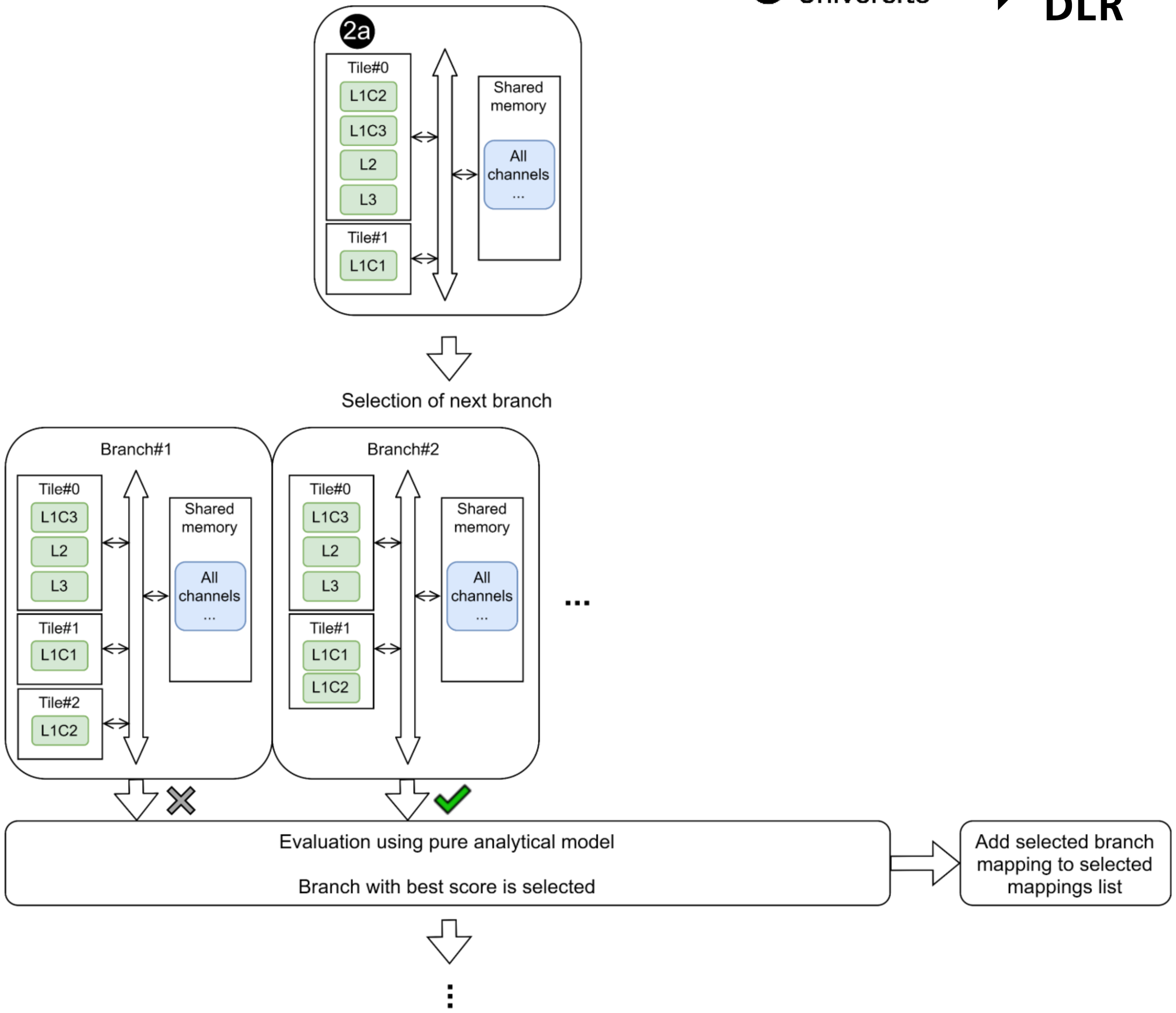
V. DSE – Branch & Bound enhanced mapping search





V. DSE – Branch & Bound enhanced mapping search





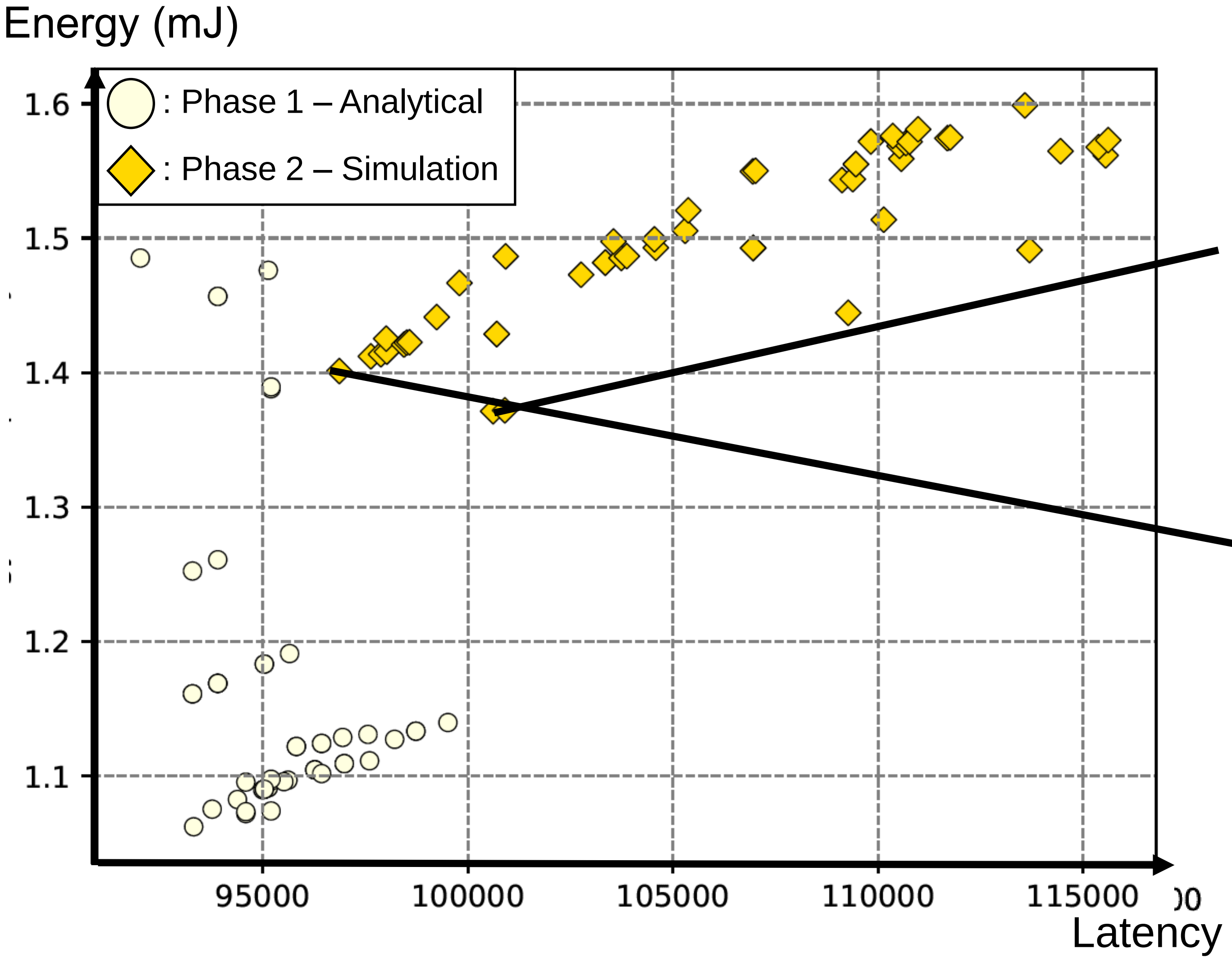
Selection of next branch

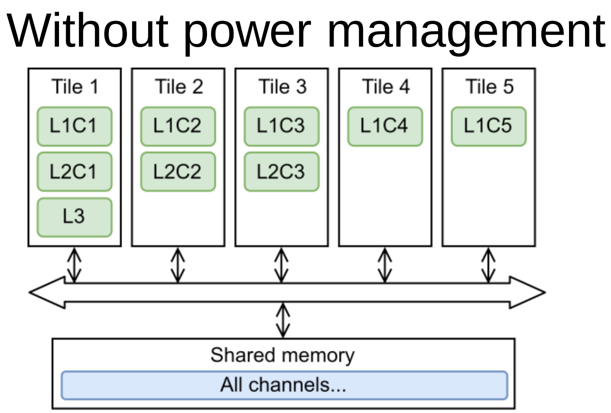
Evaluation using pure analytical model

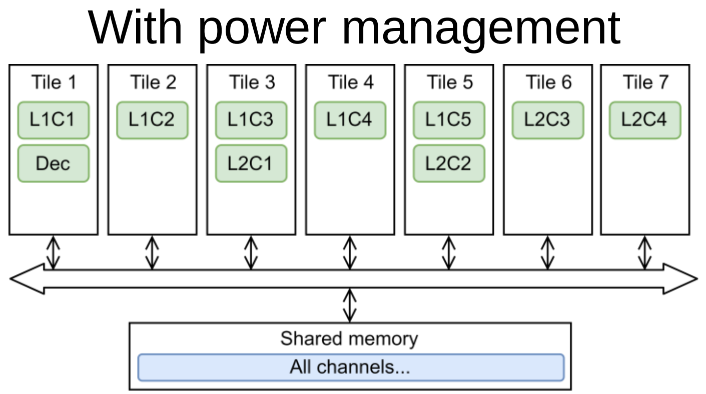
Branch with best score is selected

Add selected branch mapping to selected mappings list

V. DSE – Example use of the DSE flow





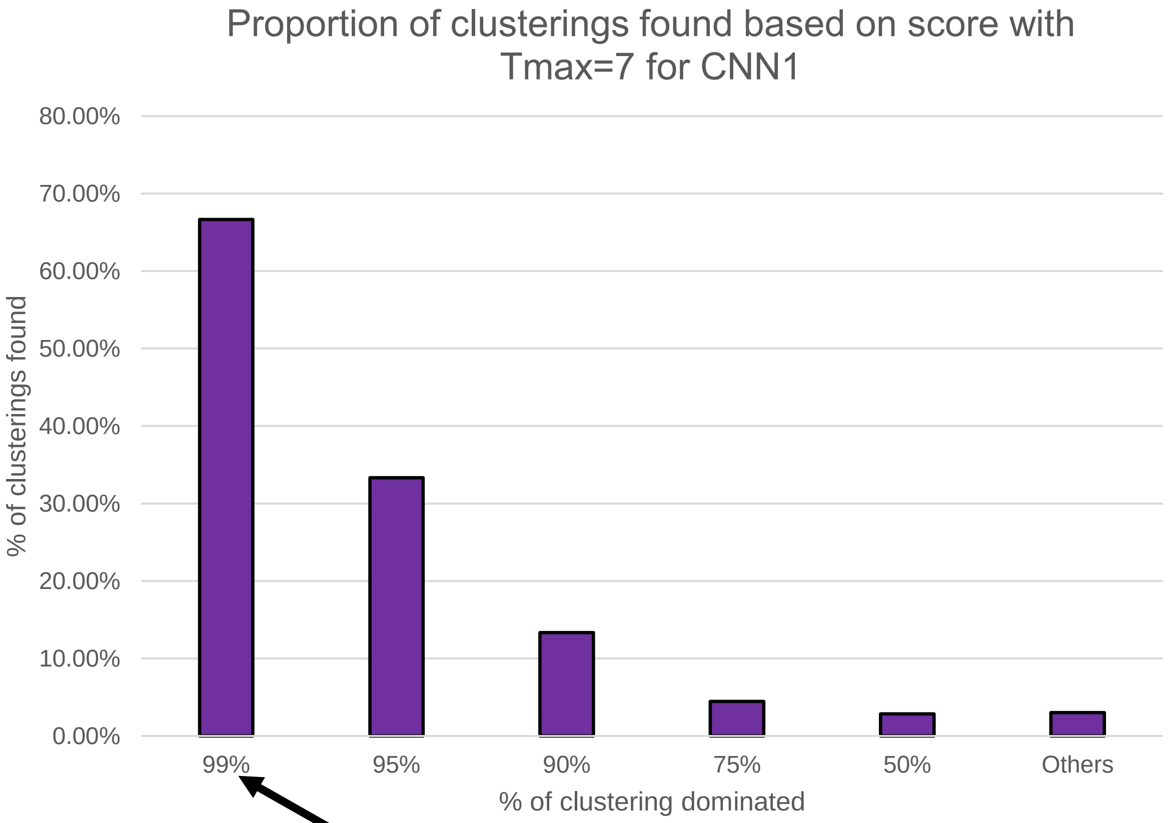


Found candidate solutions for MLP1
-> Zoom on 50 best

- The flow finds optimized non trivial solutions.
- The flow indicates when power management is worth using to enhance timing and energy.

V. DSE – Evaluation of the DSE flow

- Comparison of Branch & Bound-enhanced and exhaustive clustering search:

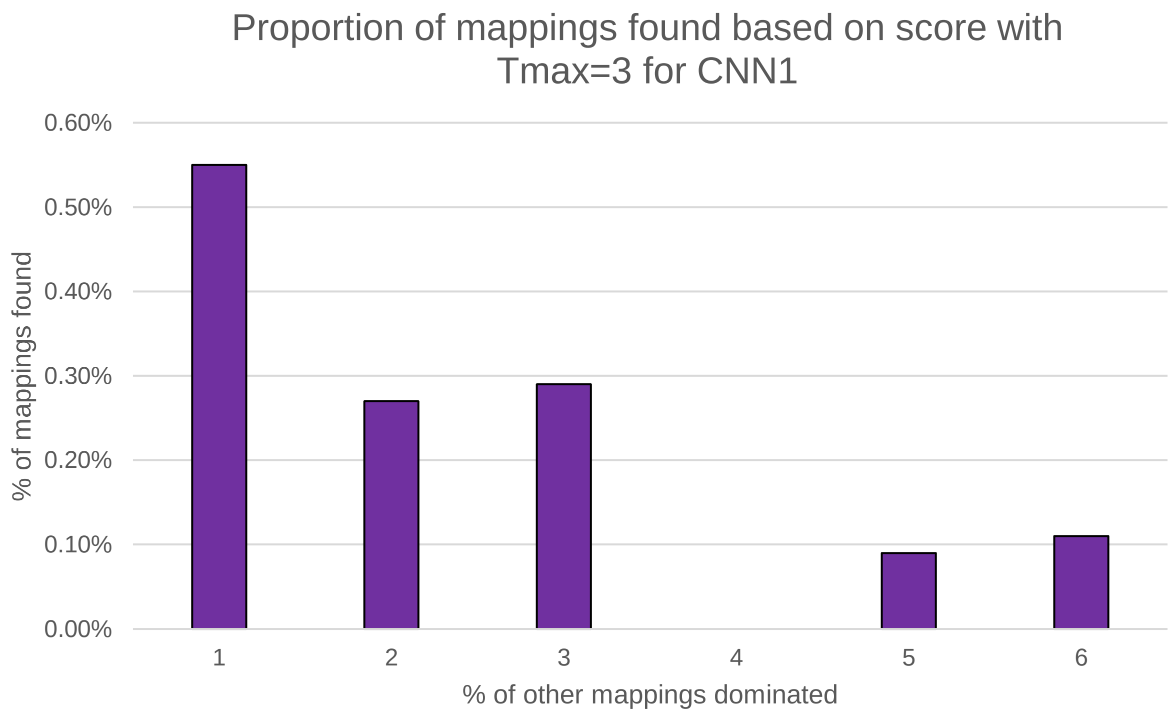


=> The optimal clustering compared to exhaustive search is always found.

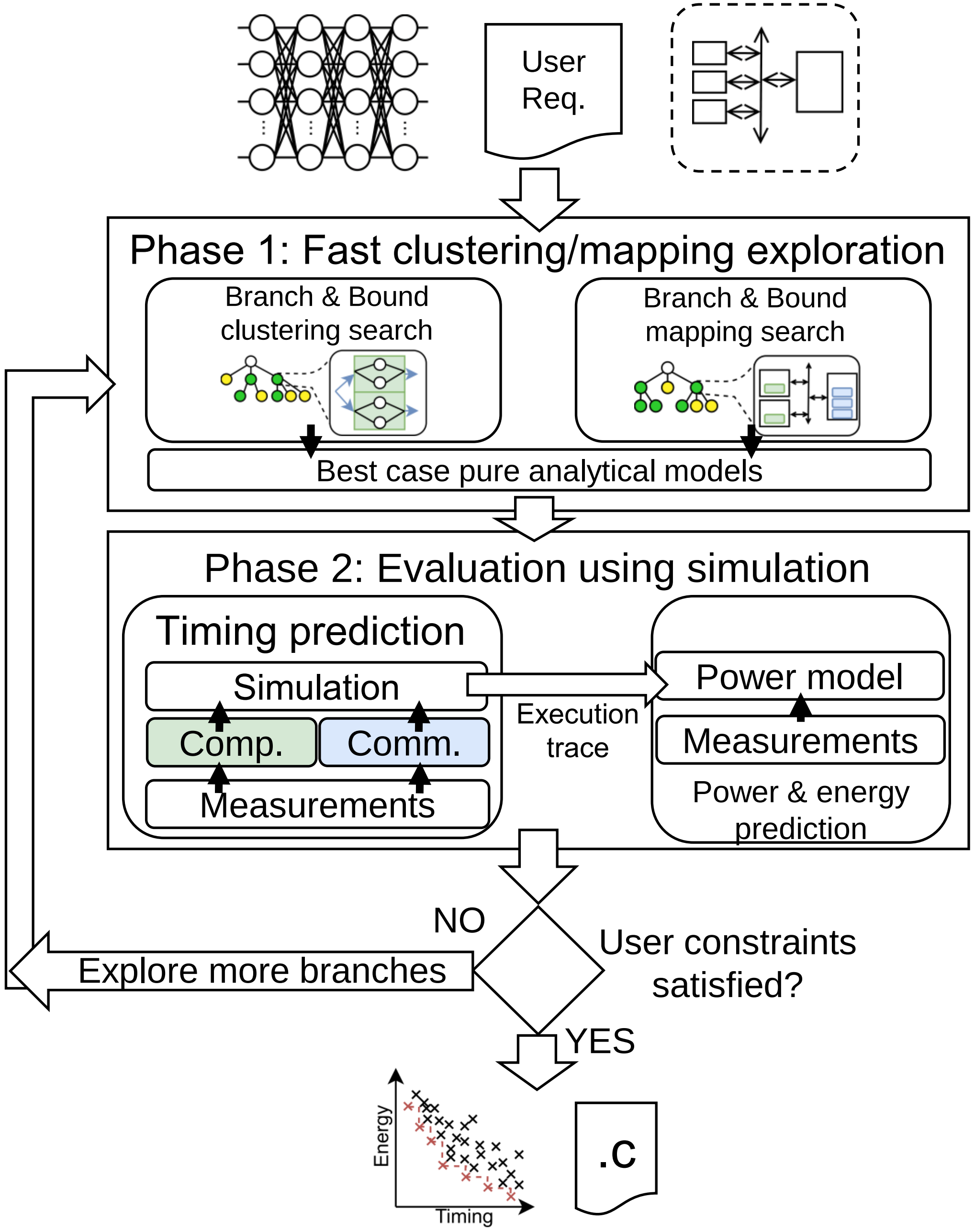
These clusterings have a better score than 99% of other clusterings found with exhaustive.

V. DSE – Evaluation of the DSE flow

- Comparison of Branch & Bound-enhanced and exhaustive mapping search
  - Similar observations. However optimal mapping is not guaranteed to be found.



- Use of pure analytical models for pruning vs simulation
  - => Similar results are obtained with the analytical models / simulation.



I. Related work overview

II. Technical background

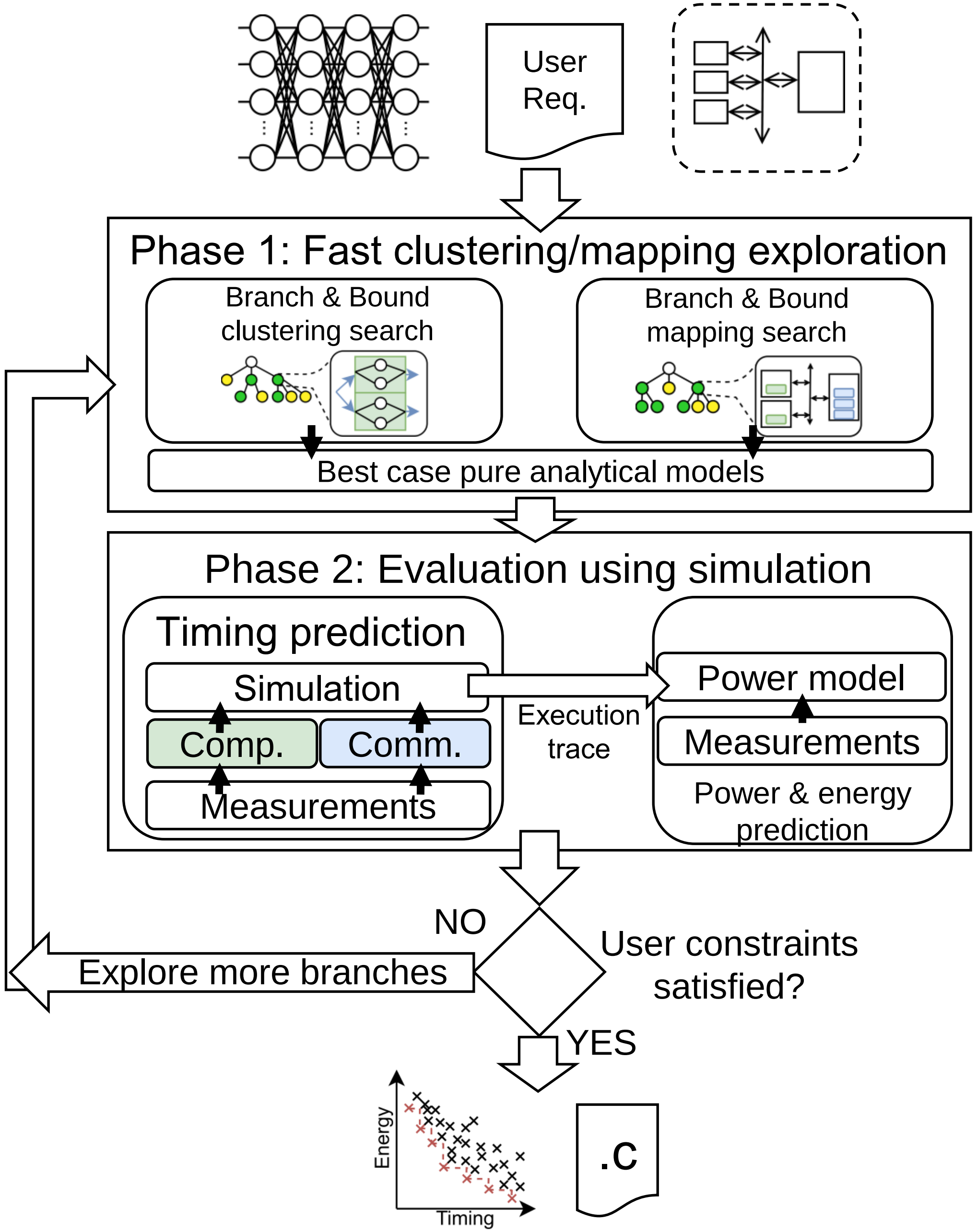
III. Timing prediction flow

IV. Power and energy analysis flow

V. Design Space Exploration flow

VI. Conclusion & Prospects

VI. Conclusion – Research challenges

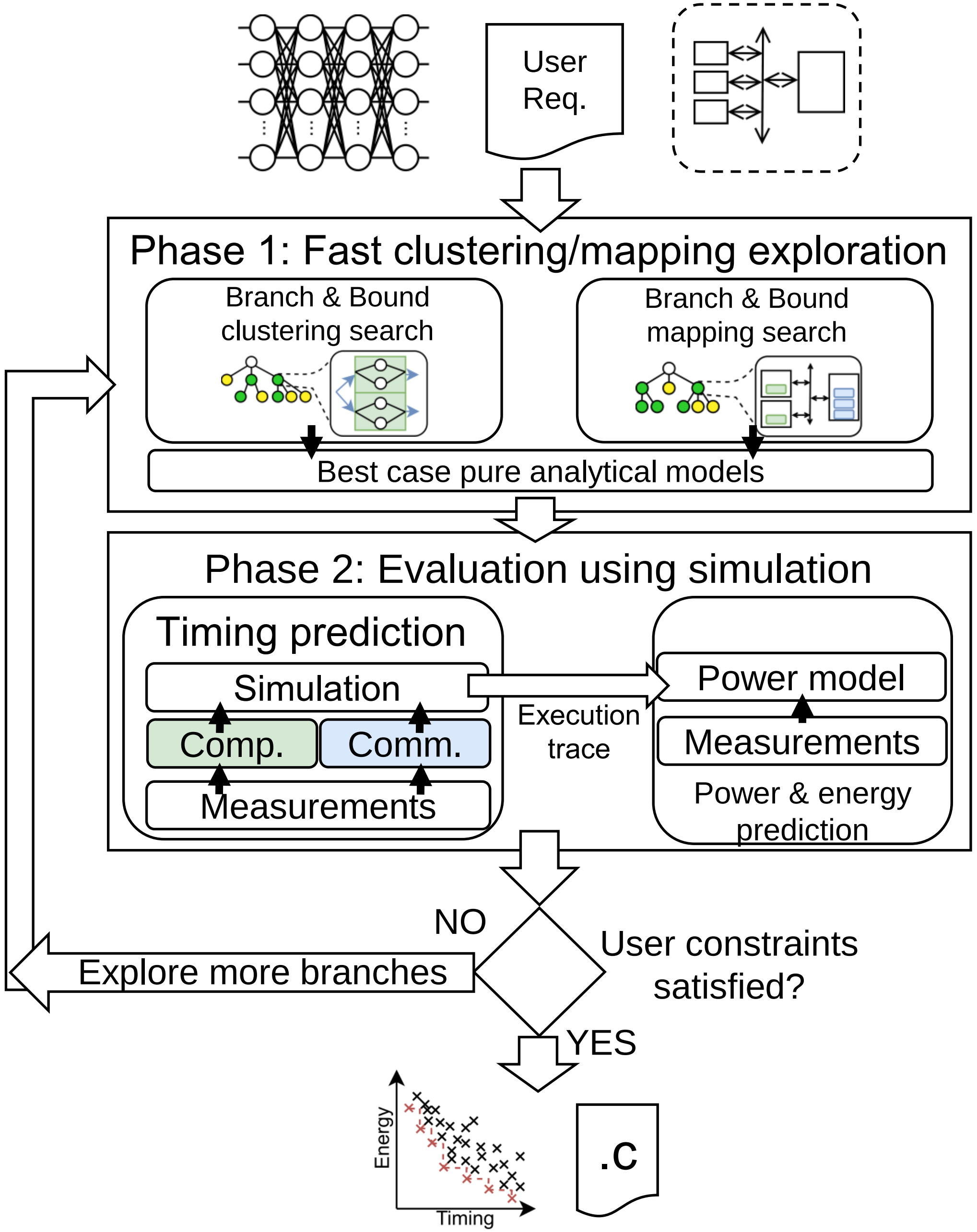

User Req.
Phase 1: Fast clustering/mapping exploration
Branch & Bound clustering search
Branch & Bound mapping search
Best case pure analytical models
Phase 2: Evaluation using simulation
Timing prediction
Simulation
Comp.
Comm.
Measurements
Execution trace
Power model
Measurements
Power & energy prediction
NO
User constraints satisfied?
Explore more branches
YES
Energy
Timing
.C

- 1. How to provide fast yet accurate evaluation early in design phases of timing and energy properties for streaming NNs deployments on multi-core platforms?
  - ➢ Use hybrid modeling flow: simulation, analytical models, measurements.

- 2. Is a model-based approach more relevant than rapid prototyping?
  - ➢ Yes. 6 times faster with high accuracy + doesn't need the NN to be trained.

- 3. Is a model-based approach suited for early, fast and confident Design Space Exploration (DSE) of streaming NNs deployments on multi-core platforms?
  - ➢ Yes, we demonstrated it with our DSE approach.

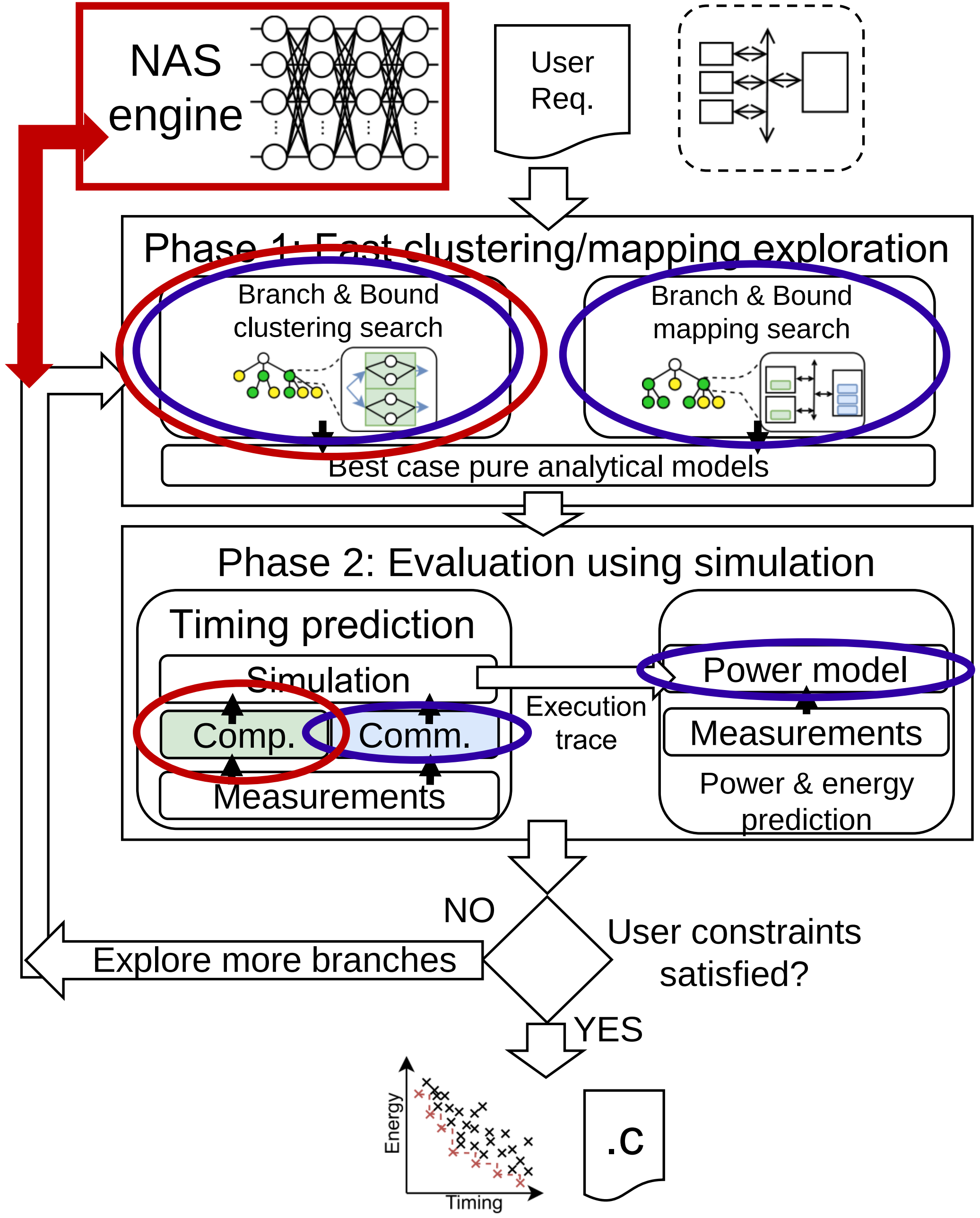
VI. Conclusion – Limitations



User Req.
Phase 1: Fast clustering/mapping exploration
Branch & Bound clustering search
Branch & Bound mapping search
Best case pure analytical models
Phase 2: Evaluation using simulation
Timing prediction
Simulation
Comp.
Comm.
Measurements
Execution trace
Power model
Measurements
Power & energy prediction
NO
Explore more branches
User constraints satisfied?
YES
Energy
Timing
.c

- Prediction error (standard deviation) on power and energy raises up to 7% with the communication rate per tile (70%).
- On single-core platforms with important private memory allocated (1024kB, 2048kB), power and energy modeling has error > 10%.
- The analytical models used for the DSE flow could be improved.

VI. Conclusion – Perspectives


NAS engine
User Req.
Phase 1: Fast clustering/mapping exploration
Branch & Bound clustering search
Branch & Bound mapping search
Best case pure analytical models
Phase 2: Evaluation using simulation
Timing prediction
Simulation
Comp.
Comm.
Measurements
Execution trace
Power model
Measurements
Power & energy prediction
NO
Explore more branches
User constraints satisfied?
YES
Energy
Timing
.c

- Extend the flow to support Neural Architecture Search (NAS) [1]
- Offer modeling and exploration of external memory accesses (necessary for larger NNs)

[1] Elsken, T.; Metzen, J. H. & Hutter, F. "Neural Architecture Search: A Survey“, *Journal of Machine Learning Research (JMLR)*, **2019**

Quentin Dariol’s PhD defense – 27.11.2023

Early Timing and Energy Prediction and Optimization of Artificial Neural Networks on Multi-Core Platforms

QUESTIONS

Evaluation committee:

- Dr. Kim GRÜTTNER
- Prof. Dr. Matthias JUNG
- Dr. Angeliki KRITIKAKOU
- Prof. Dr. Frédéric PÉTROT
- Prof. Dr. Gregor SCHIELE

Guest:

- Dr. Domenik HELMS

Reviewers before defense:

- Prof. Dr. Matthias JUNG
- Dr. Angeliki KRITIKAKOU

Thesis work supervised by:

- Prof. Dr. Sébastien PILLEMENT
- Dr. Sébastien LE NOURS

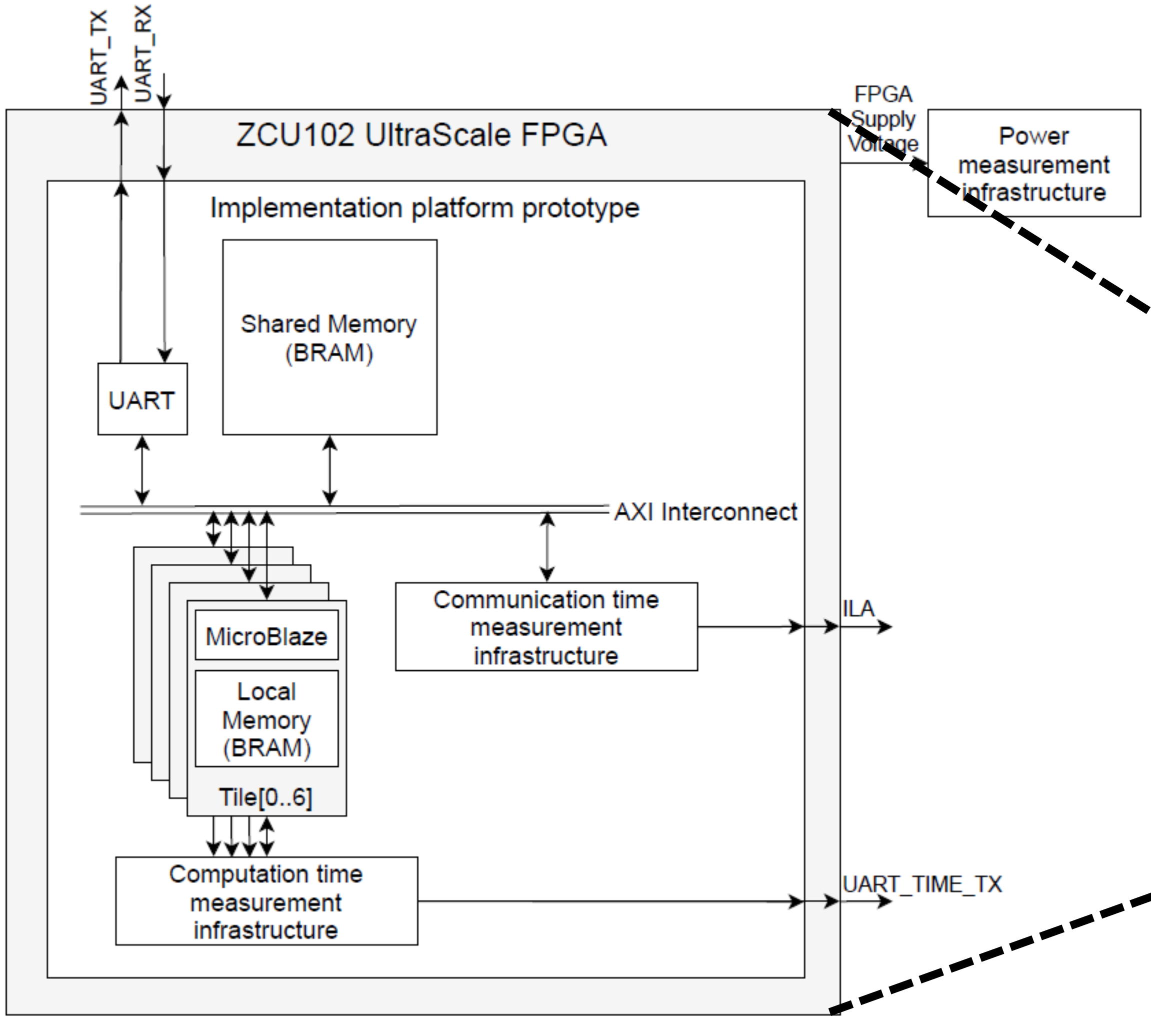
IETR

Nantes Université

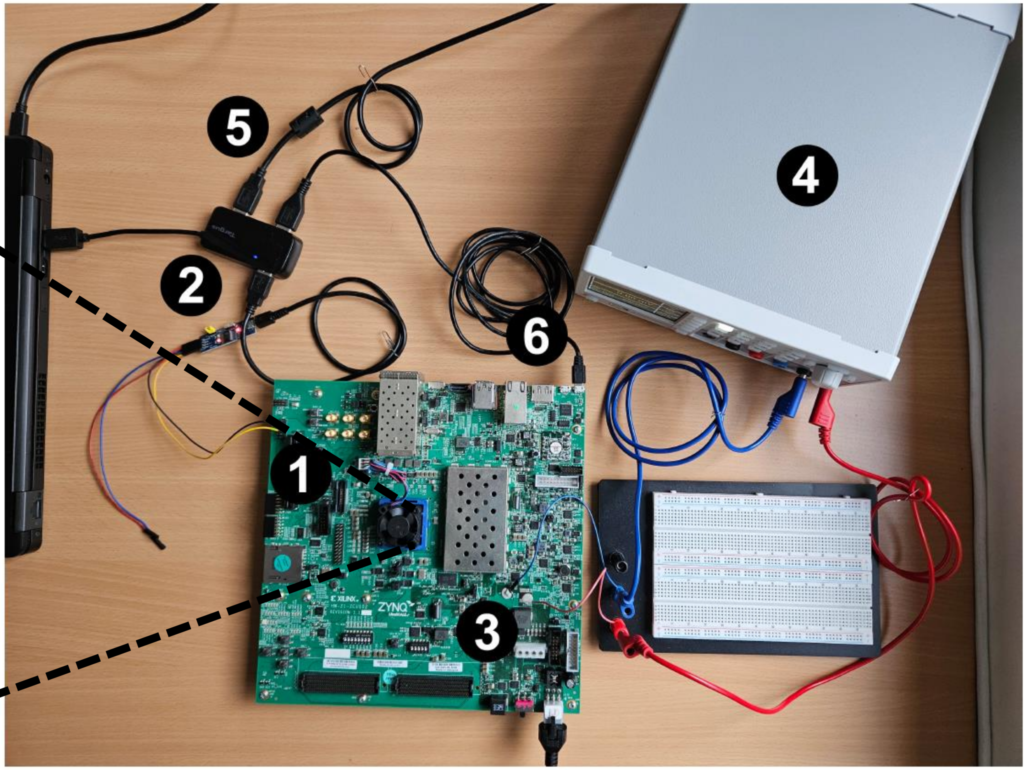
DLR

APPENDICES

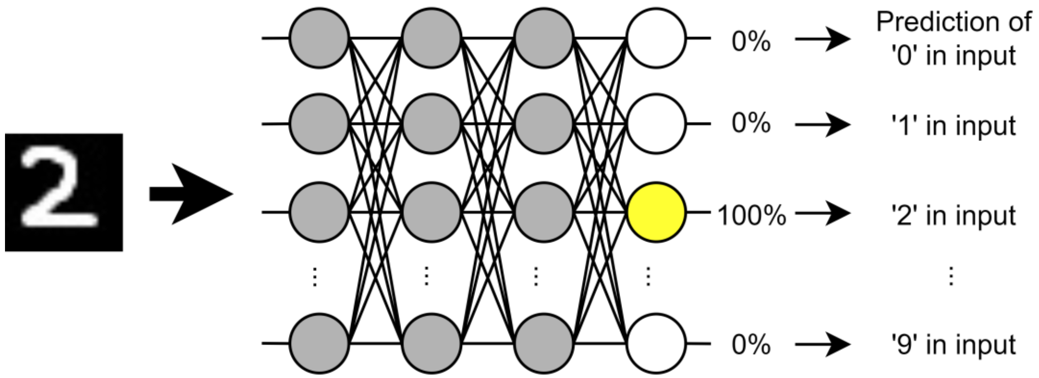
Appendice – Prototype platform



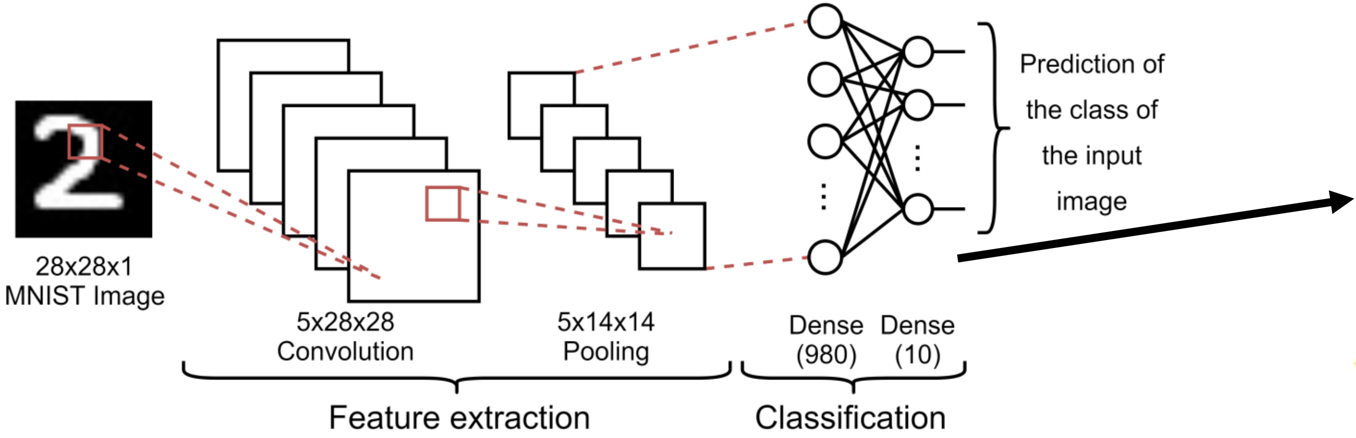
UART_TX
UART_RX
ZCU102 UltraScale FPGA
FPGA Supply Voltage
Power measurement infrastructure
Implementation platform prototype
Shared Memory (BRAM)
UART
AXI Interconnect
MicroBlaze
Local Memory (BRAM)
Tile[0..6]
Communication time measurement infrastructure
ILA
Computation time measurement infrastructure
UART_TIME_TX

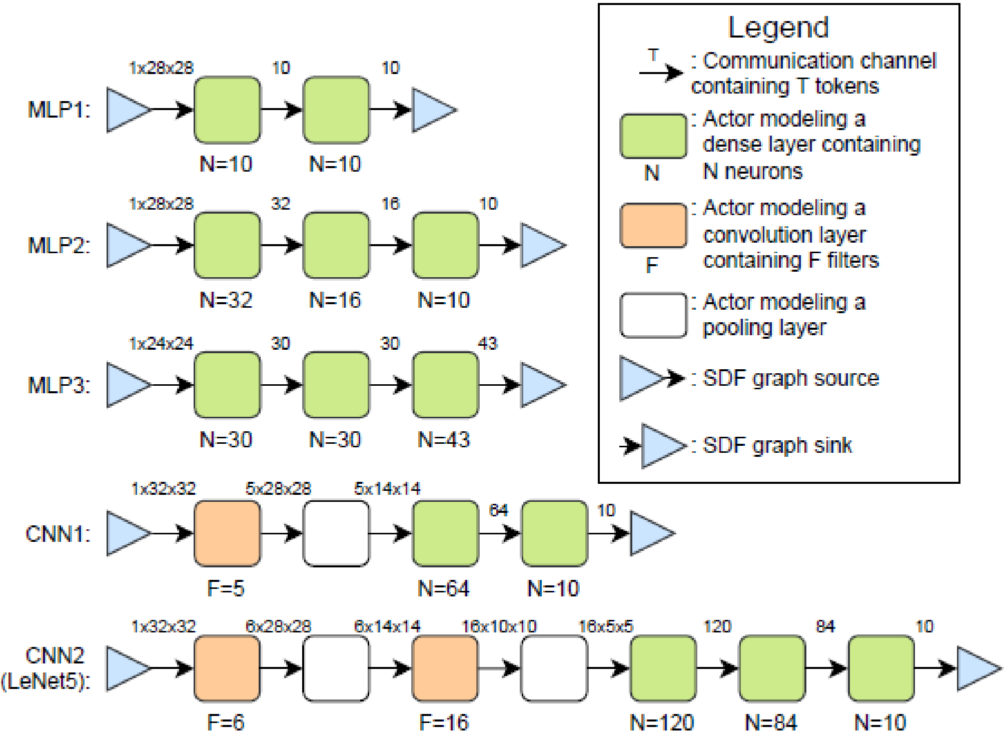


I. Fundamentals & hypothesis – Considered NNs

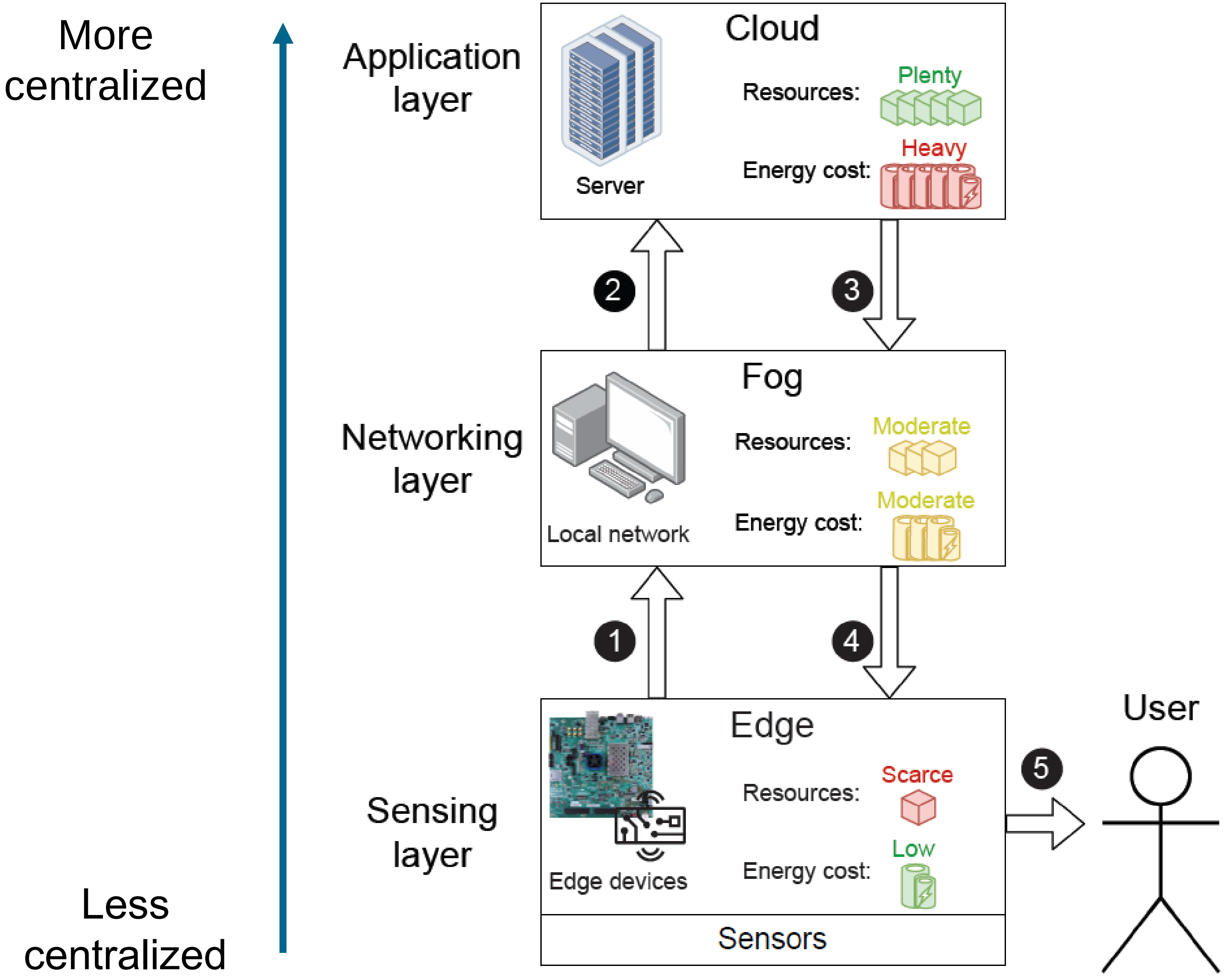


NN name	Number of layers	Data-set	Accuracy
MLP1	2	MNIST [11]	85%
MLP2	3	MNIST [11]	89%
MLP3	3	GTSRB [90]	20%
CNN1	4	MNIST [11]	77%
CNN2	7	MNIST [11]	N.A.





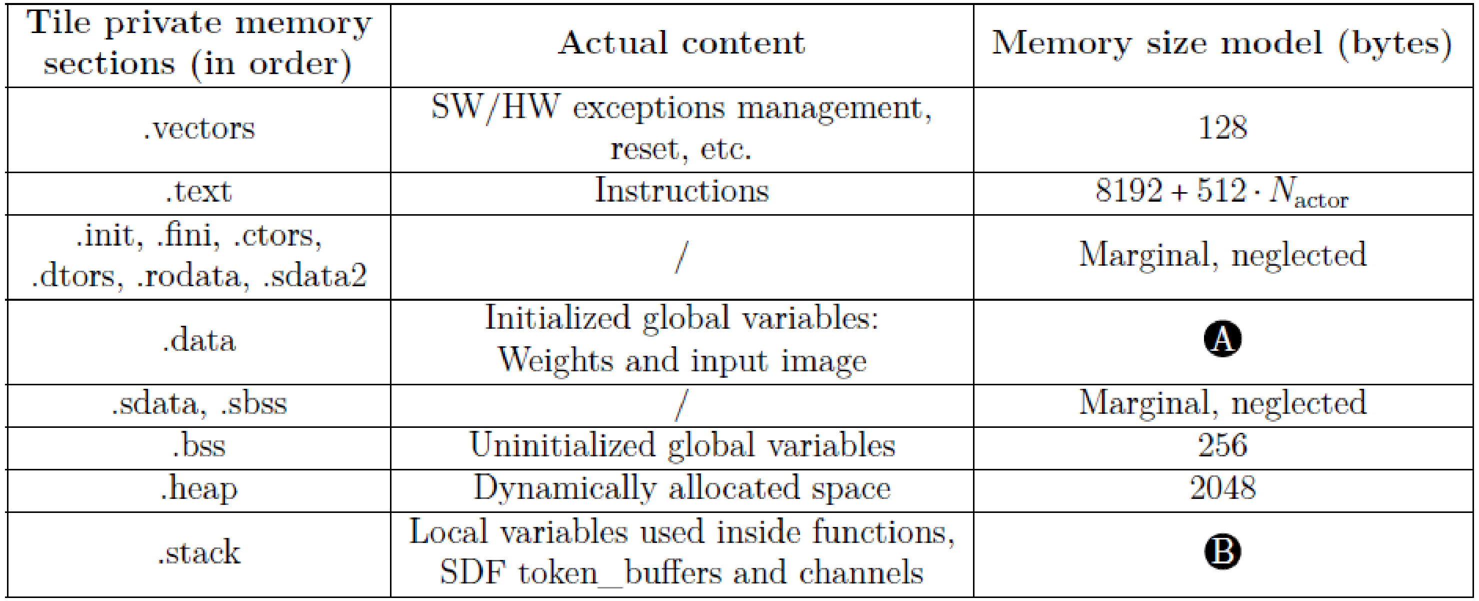
Context – Internet of Things (IoT)



Inspiration for figure:

- Gunathilake, N. A.; Buchanan, W. J. & Asif, R. "Next Generation Lightweight Cryptography for Smart IoT Devices: Implementation, Challenges and Applications« , *2019 IEEE 5th World Forum on Internet of Things (WF-IoT),* 2019
- ur Rehman, M. H.; Yaqoob, I.; Salah, K.; Imran, M.; Jayaraman, P. P. & Perera, C., "The role of big data analytics in industrial Internet of Things“, *Future Generation Computer Systems,* 2019

Appendice – Private memory model for tile sizing



Tile private memory sections (in order)	Actual content	Memory size model (bytes)
.vectors	SW/HW exceptions management, reset, etc.	128
.text	Instructions	$8192 + 512 \cdot N_{\text{actor}}$
.init, .fini, .ctors, .dtors, .rodata, .sdata2	/	Marginal, neglected
.data	Initialized global variables: Weights and input image	🅐
.sdata, .sbss	/	Marginal, neglected
.bss	Uninitialized global variables	256
.heap	Dynamically allocated space	2048
.stack	Local variables used inside functions, SDF token_buffers and channels	🅑

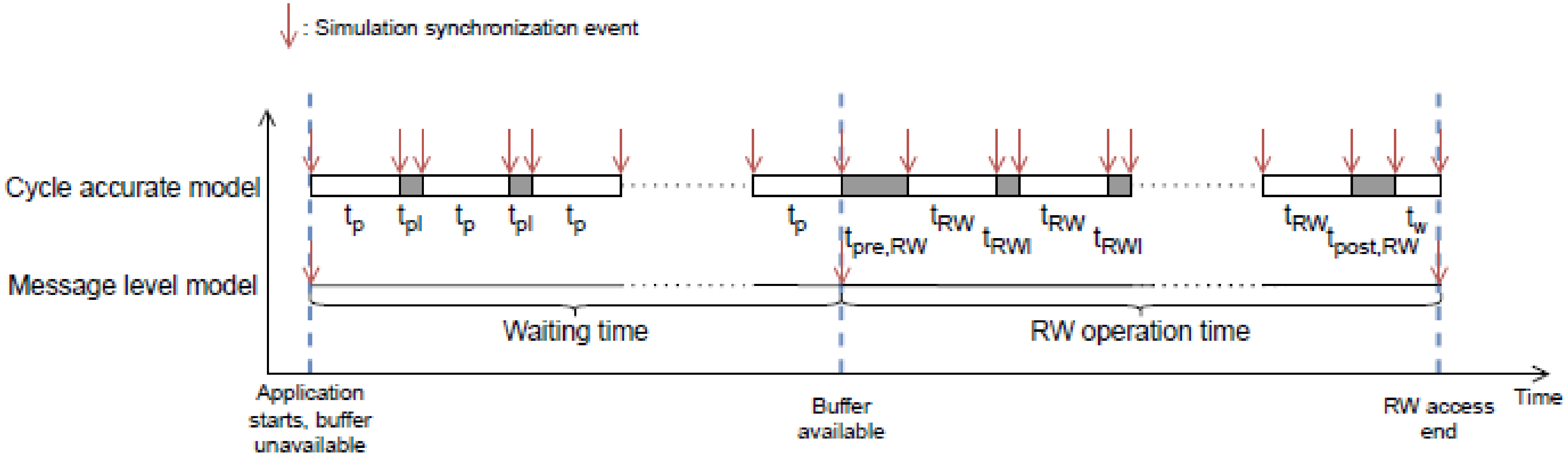
$$🅐 : B_{l=0} + \sum_{a=1}^{A} \lambda_a W_a$$

$$\text{with} \quad W_a = \begin{cases} 4 \cdot N_{\text{neuron},a} \cdot (N_{\text{inputs}} + 1) & \text{if } a \text{ is an actor from a dense layer} \\ 4 \cdot K_a \cdot (F_{\text{h}} \cdot F_{\text{w}} + N_{\text{inputs}}) & \text{if } a \text{ is an actor from a convolution layer} \end{cases}$$

$$🅑 : 4 \cdot \left(2 \cdot N_{\text{channels}} + \sum_{l=0}^{L-1} B_l \right)$$

$$\text{and} \quad \lambda_a = \begin{cases} 1 & \text{if actor } a \text{ is mapped on the considered tile} \\ 0 & \text{otherwise} \end{cases}$$

Appendice – Communication time model



$$D_{RW}(n_T) = \underbrace{t_{init,RW} + t_p}_{\text{Check token availability}} \underbrace{+t_{pre,RW} + t_{RW} \cdot n_T + t_{RWl} \cdot (n_T - 1)}_{\text{Buffer access}} \underbrace{+t_{post,RW} + t_w}_{\text{Token status update}}$$

Communication procedure	t_r	t_p	t_w	t_{rl}	t_{wl}	t_{pl}	$t_{r_{loop}}$	$t_{w_{loop}}$	$t_{p_{loop}}$	t_{pr_r}	t_{po_r}	t_{pr_w}	t_{po_w}	t_{init_r}	t_{init_w}
Polling	8	**8**	5	14	13	**7**	22	18	**15**	15	11	15	9	**15**	**16**
Interrupt	8	**0**	5	14	13	**0**	22	18	**0**	15	11	15	9	**348**	**349**

*: All delays in processor cycles