Formal Modeling and Schedulability Analysis of HPC-DAGs Using Time Petri Nets

KHADIRI Houda[1,2] BOUKIR Khaoula[1] HLADIK Pierre-Emmanuel[2] ZAHAF Houssam-Eddine[2]

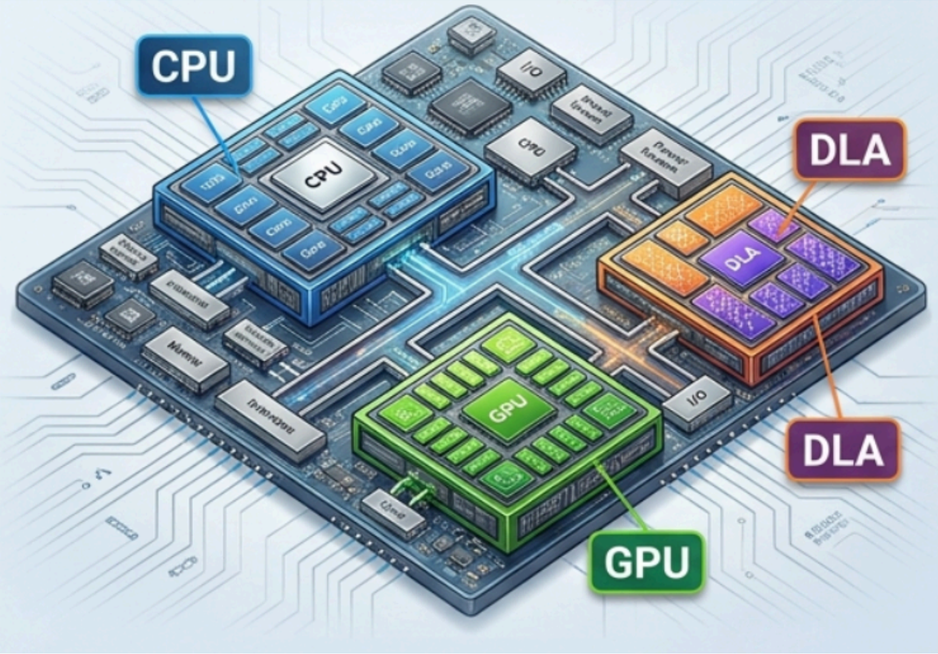
Université Ibn Tofail, SETIME[1] – Nantes Université, École Centrale Nantes, LS2N [2]

07 - septembre - 2026

1. CONTEXTE ET MOTIVATION

1. CONTEXTE ET MOTIVATION

Plateformes modernes

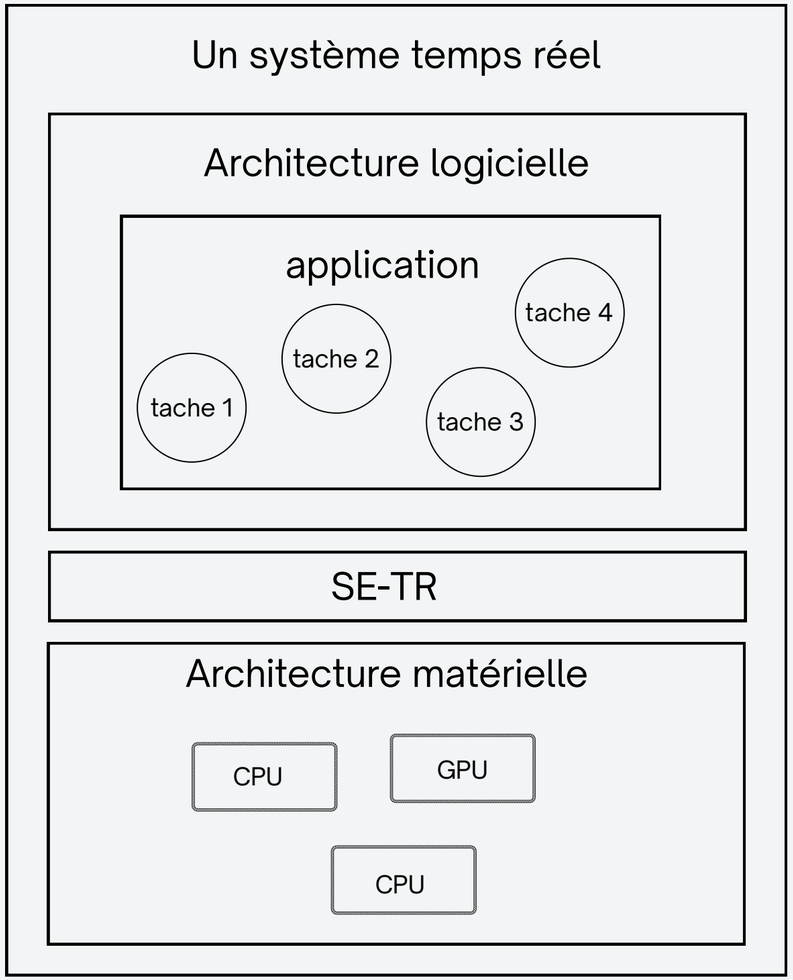


Les systèmes cyber-physiques (robotique, controle industriel ...) intègrent désormais des architectures hautement hétérogènes.

L'exigence temporelle

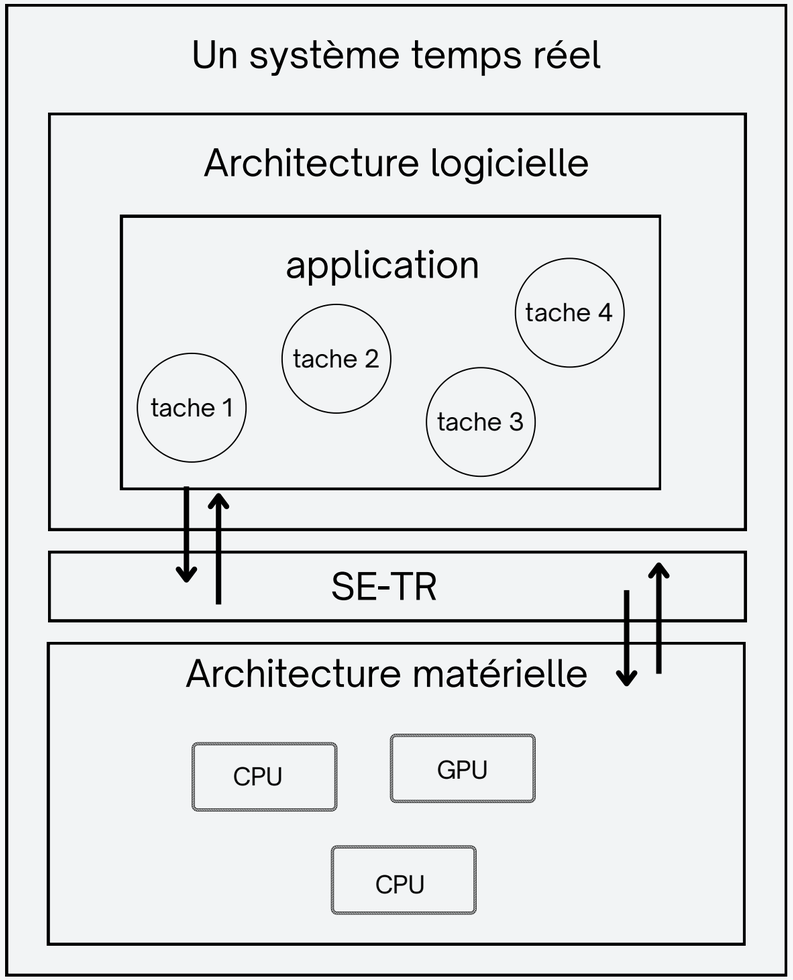
Ces systèmes nécessitent des garanties strictes . L'analyse d'ordonnaçabilité est fondamentale pour prouver qu'aucune tâches ne dépassera son délai

ANALYSE D'ORDONNANÇABILITÉ



Est ce que l'architecture matérielle est capable d'exécuter l'architecture logicielle en respectant les contraintes temporelles?

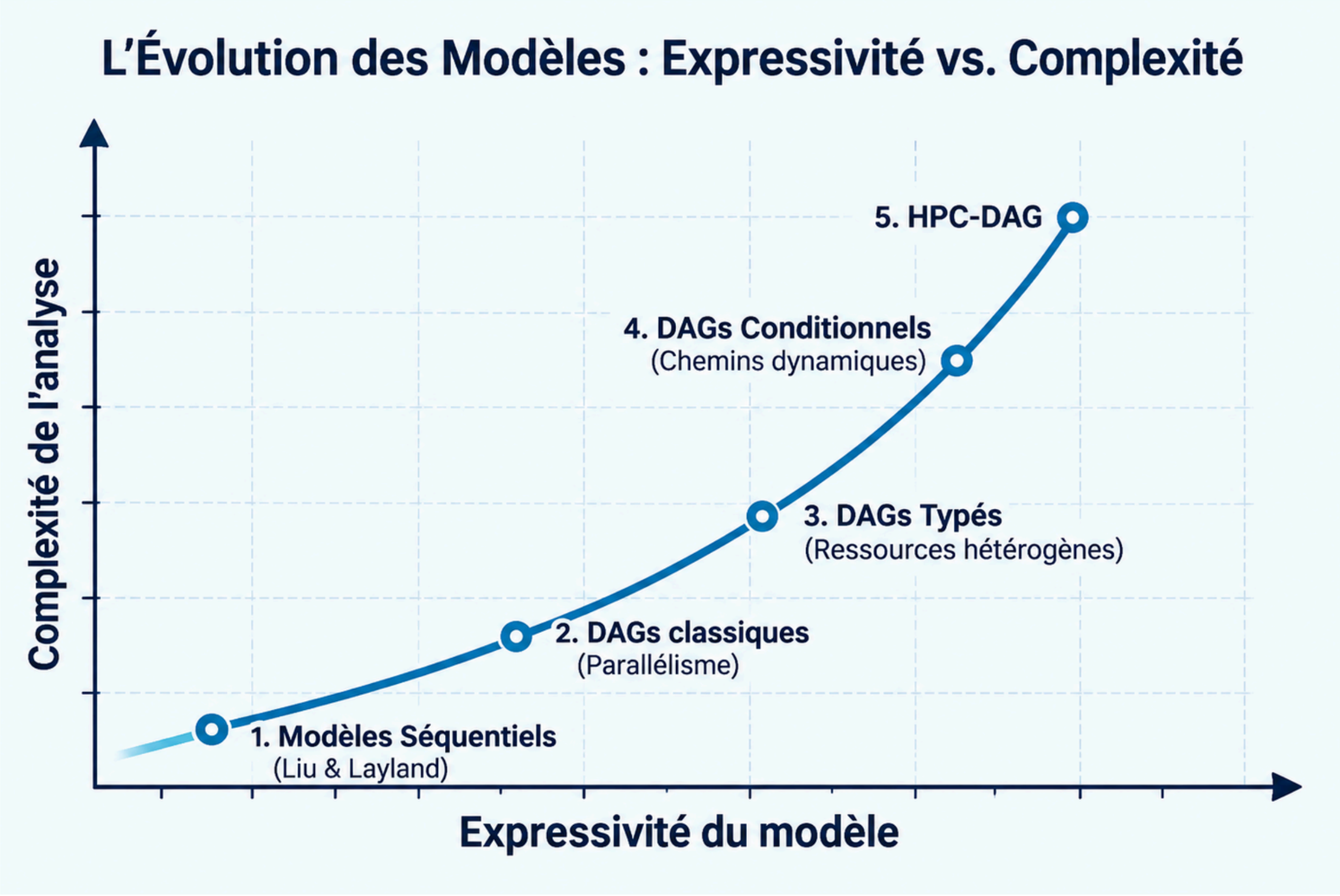
ANALYSE D'ORDONNANÇABILITÉ



Est ce que l'architecture matérielle est capable d'exécuter l'architecture logicielle en respectant les contraintes temporelles?

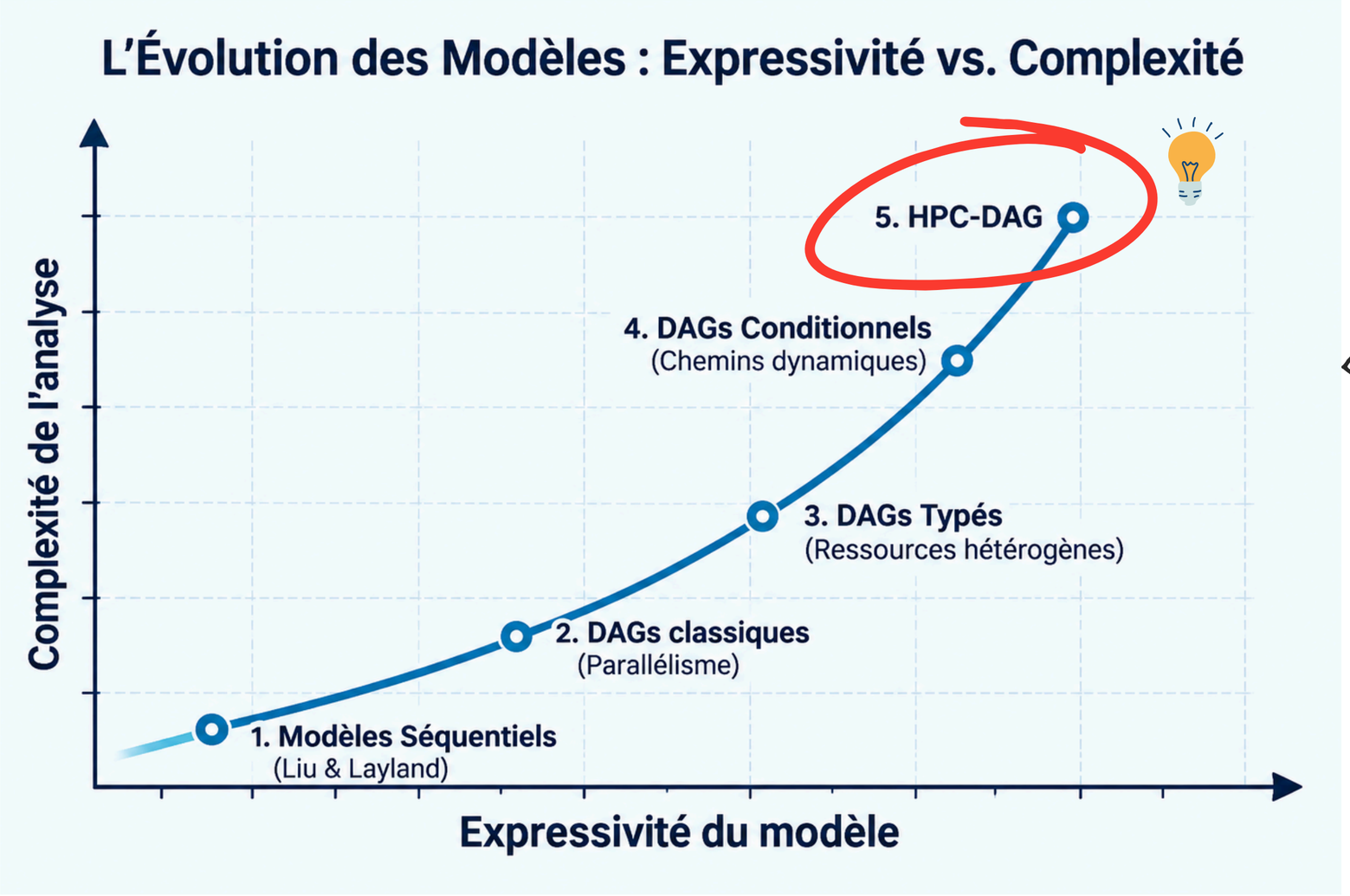
Raisonnement sur un modèle mathémathique

MODÉLISATION MATHÉMATIQUE



Les modèles simples ne reflètent plus la réalité matérielle

MODÉLISATION MATHÉMATIQUE



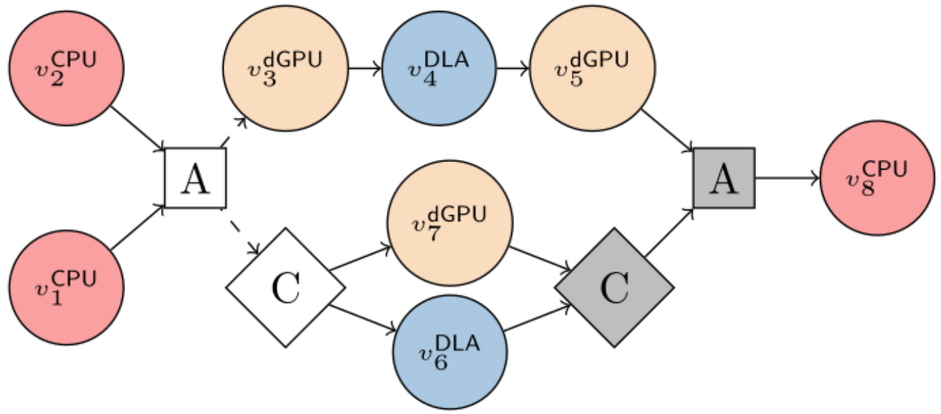
Les modèles simples ne reflètent plus la réalité matérielle

MODÉLISATION MATHÉMATIQUE
ARCHITECTURE HETEROGENE

plateformes avec différents types d'architectures :
ex : (8 CPUs, 2 GPUs, 2 DLAs)

HPC-DAG Model : (Heterogeneous Parallel Condition Directed Acyclic Graph)

$\tau = \tau_1, \tau_2, ..., \tau_n$ / $\tau = \{\mathsf{T}, \mathsf{D}, \mathcal{V}, \mathcal{A}, \Gamma, \mathcal{E}\}$



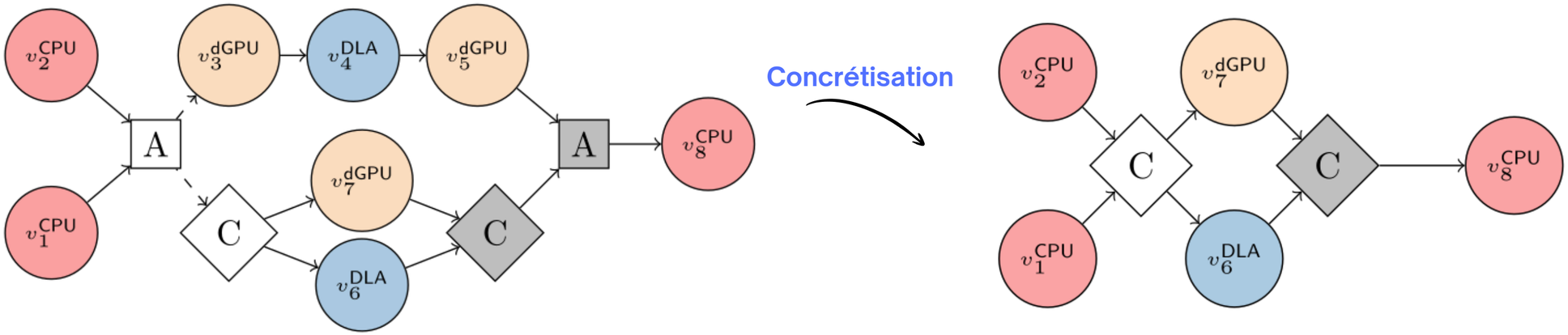
Specification task

MODÉLISATION MATHÉMATIQUE
ARCHITECTURE HETEROGENE

plateformes avec différents types d'architectures :
ex : (8 CPUs, 2 GPUs, 2 DLAs)

HPC-DAG Model :

$\tau = \tau_1, \tau_2, ..., \tau_n$ / $\tau = \{\mathsf{T}, \mathsf{D}, \mathcal{V}, \mathcal{A}, \Gamma, \mathcal{E}\}$



Specification task

Concrete task

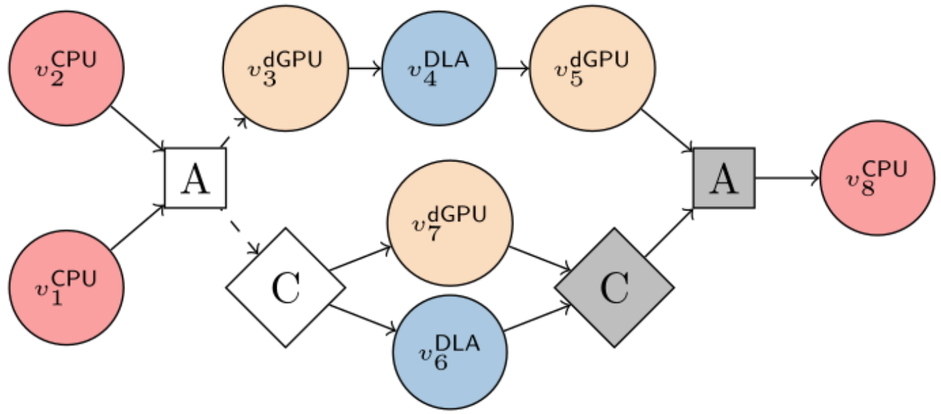
MODÉLISATION MATHÉMATIQUE
ARCHITECTURE HETEROGENE

plateformes avec différents types d'architectures :
ex : (8 CPUs, 2 GPUs, 2 DLAs)

HPC-DAG Model :

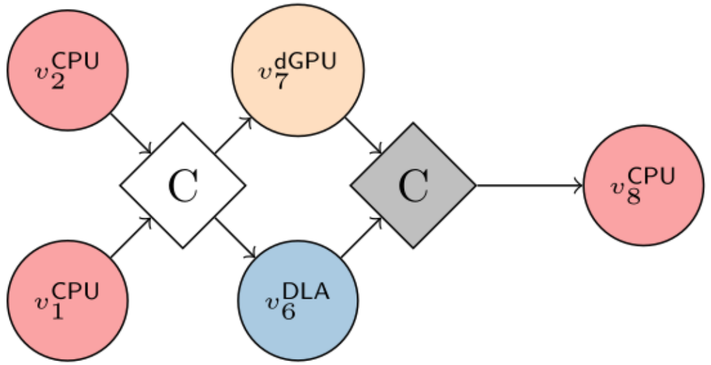
$\tau = \tau_1, \tau_2, ..., \tau_n$ / $\tau = \{\mathsf{T}, \mathsf{D}, \mathcal{V}, \mathcal{A}, \Gamma, \mathcal{E}\}$

Le Prix de l'Expressivité : Si une tâche contient k blocs avec A_i implémentations, le nombre de configurations de tâches concrètes générées explose ($\prod A_i$).



Specification task

Concrétisation



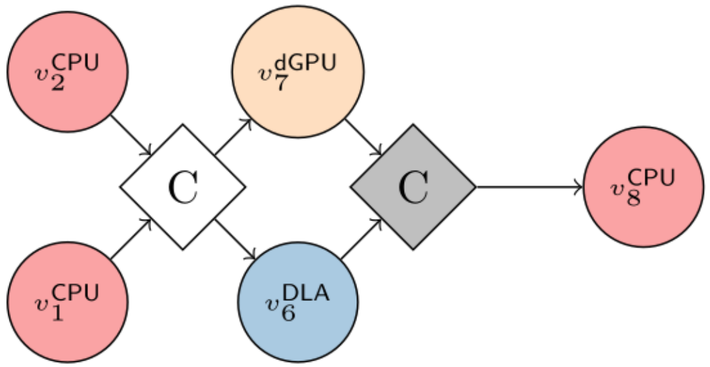
Concrete task

MODÉLISATION MATHÉMATIQUE
ARCHITECTURE HETEROGENE

plateformes avec différents types d'architectures :
ex : (8 CPUs, 2 GPUs, 2 DLAs)

HPC-DAG Model :

$\tau = \tau_1, \tau_2, ..., \tau_n$ / $\tau = \{\mathsf{T}, \mathsf{D}, \mathcal{V}, \mathcal{A}, \Gamma, \mathcal{E}\}$



Concrete task

Analyse d'ordonnancabilité

DBF (demand-bound-function) [Zahaf et al. 2021]

Pessimiste

Complexe

3. NOTRE APPROCHE

MODEL CHECKING - RPT (Réseaux de Petri temporisés)

COMMENT FONCTIONNE UN MODEL CHECKER ?

1.Modèle

Une representation formelle du systeme à verfier.

ex : RPT , automate temprisé

→

2. PROPRIÉTÉ

Une exigence à vérifier

ex : un mauvais état qui ne doit jamais arriver

→

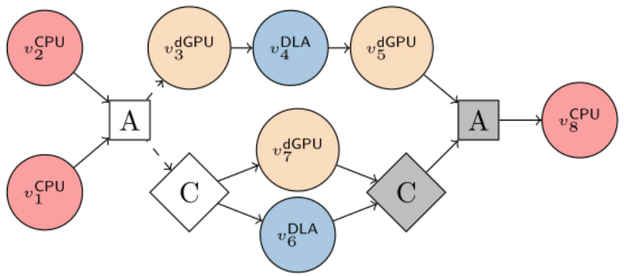
3. VÉRIFICATION

Exploration systématique de tous les états possibles

→ pas seulement un scénario

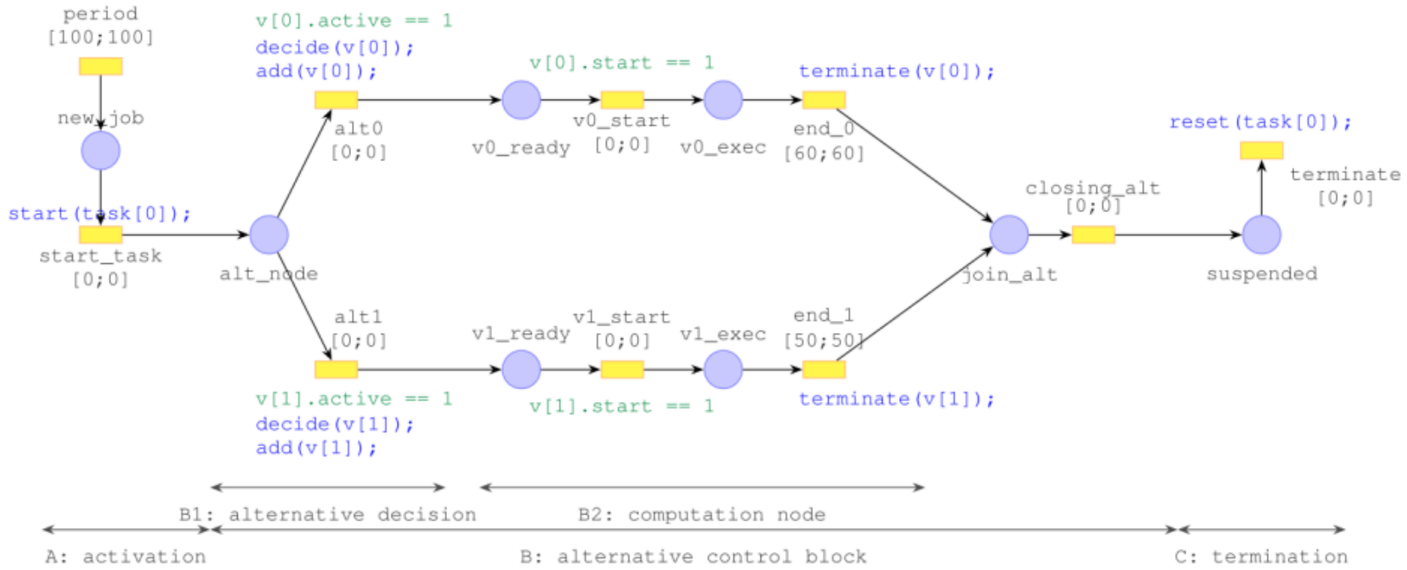
ANALYSE FORMELLE DE L'ORDONNANÇABILITÉ

MODEL CHECKING - RPT (Réseaux de Petri temporisés)



(a) Task τ

Traduction

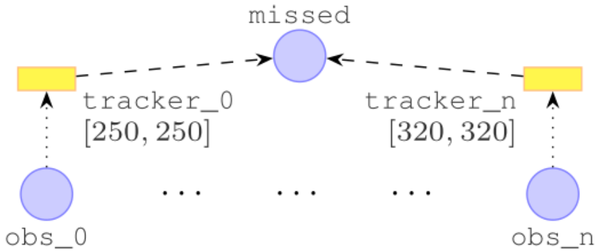


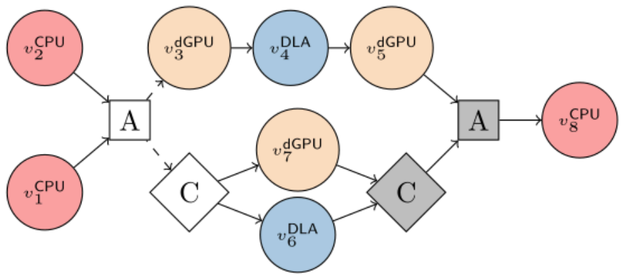
ROMEO-MODEL-CHECKER

[Lime et al. 2009]

ANALYSE FORMELLE DE L'ORDONNANÇABILITÉ

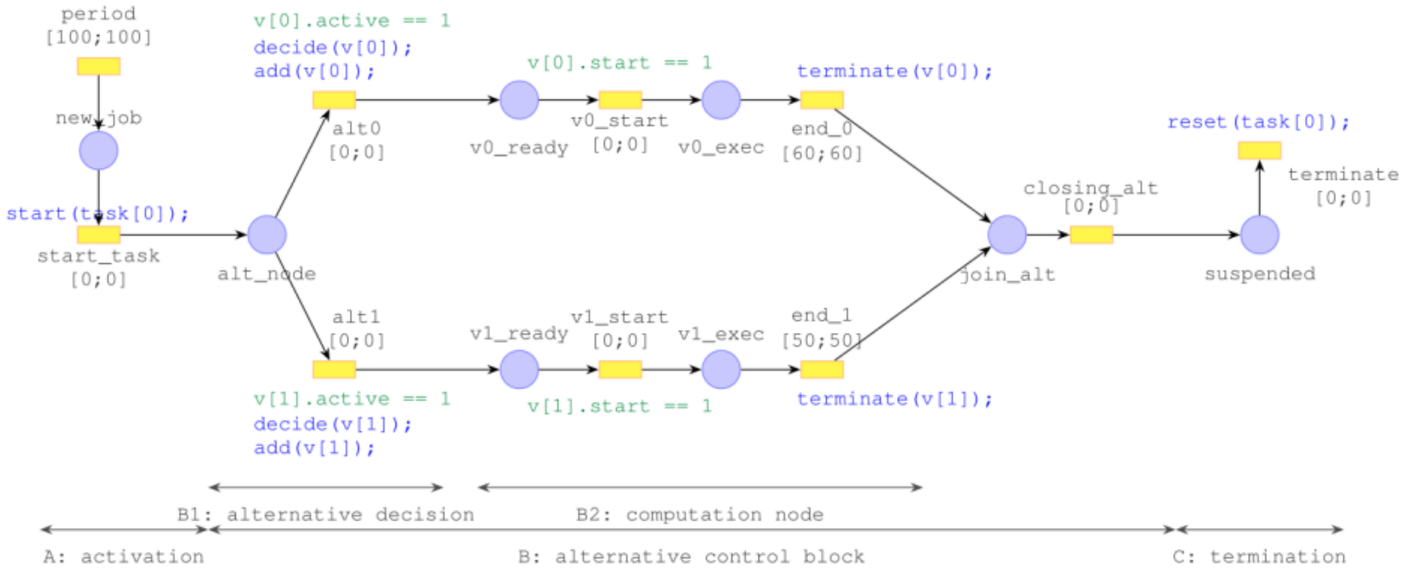
MODEL CHECKING - RPT (Réseaux de Petri temporisés)





(a) Task τ

Traduction



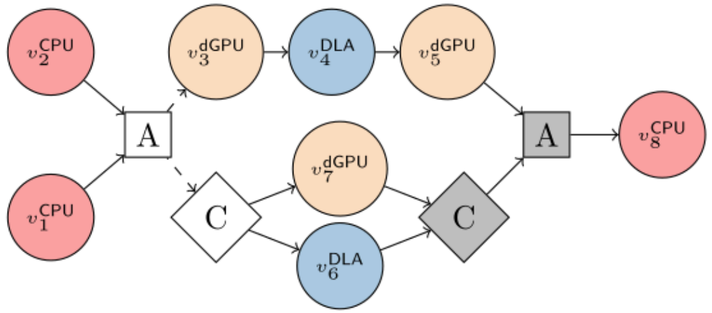
Propriété à verfier : avoid (missed > 0)
analyse d'ordonnacabilté → analyse d'atteignabilité

ROMEO-MODEL-CHECKER

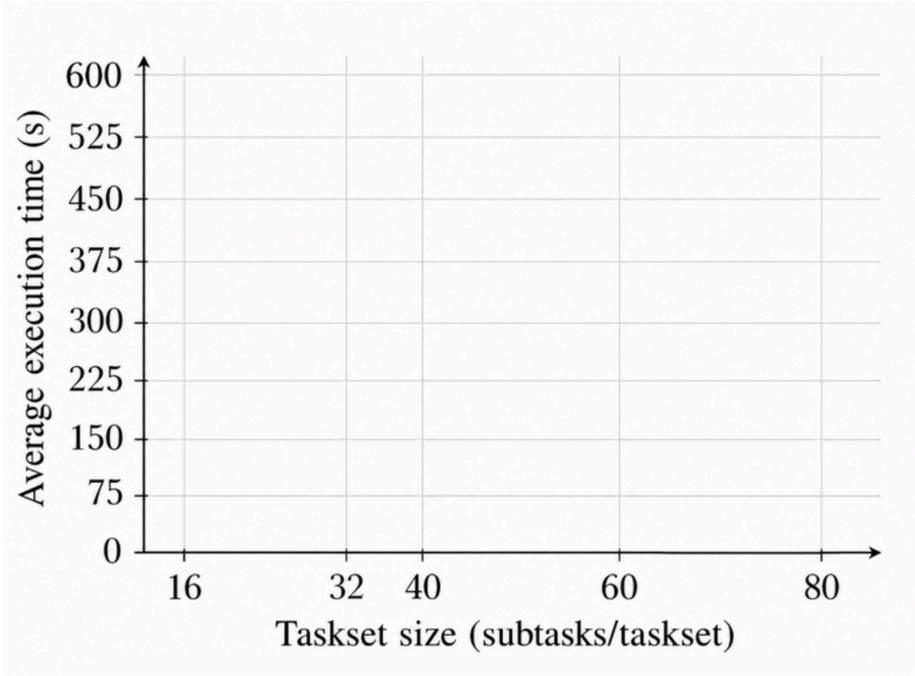
[Lime et al. 2009]

ANALYSE FORMELLE DE L'ORDONNANÇABILITÉ

MODEL CHECKING - RPT (Réseaux de Petri temporisés)

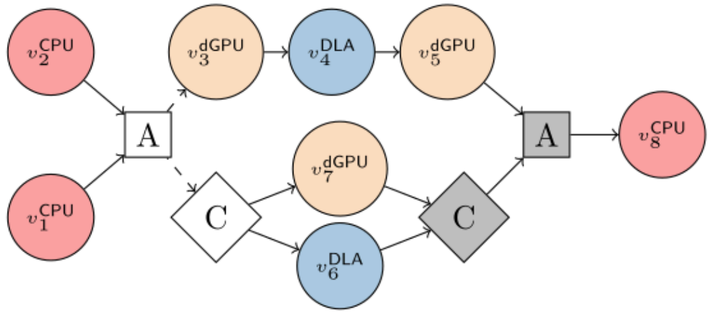


(a) Task τ

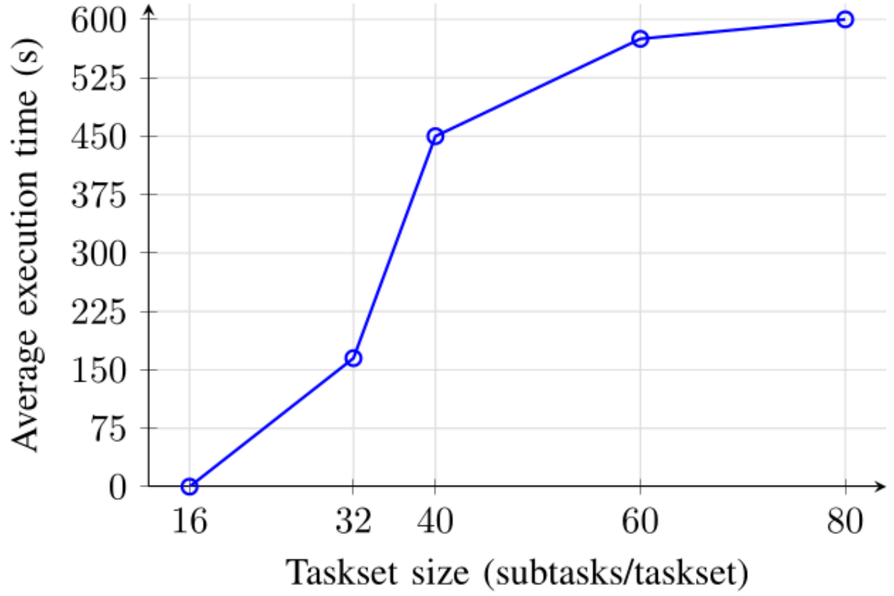


ANALYSE FORMELLE DE L'ORDONNANÇABILITÉ

MODEL CHECKING - RPT (Réseaux de Petri temporisés)

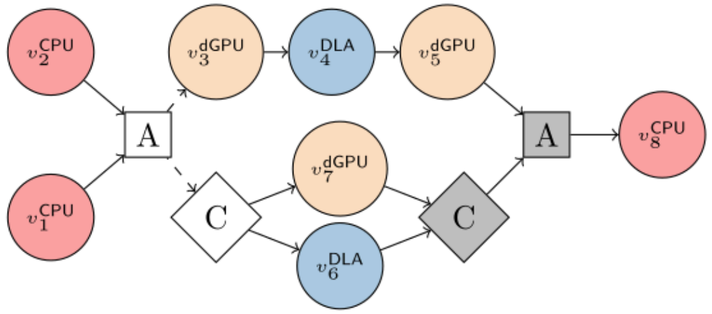


(a) Task τ



ANALYSE FORMELLE DE L'ORDONNANÇABILITÉ

MODEL CHECKING - RPT (Réseaux de Petri temporisés)

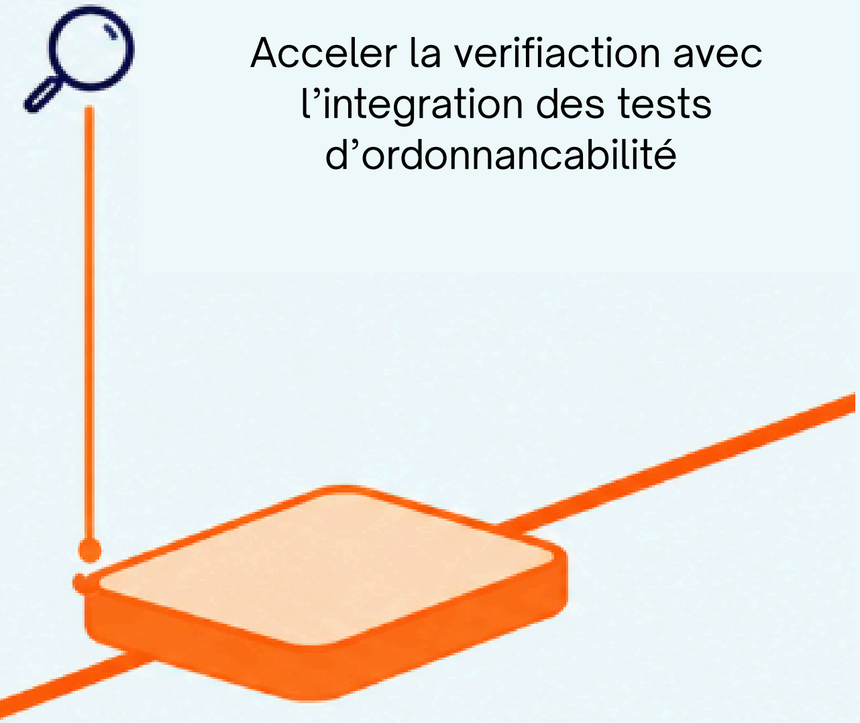


(a) Task τ

- Analyse exacte de l'ordonnançabilité
- ne passe pas à l'échelle !

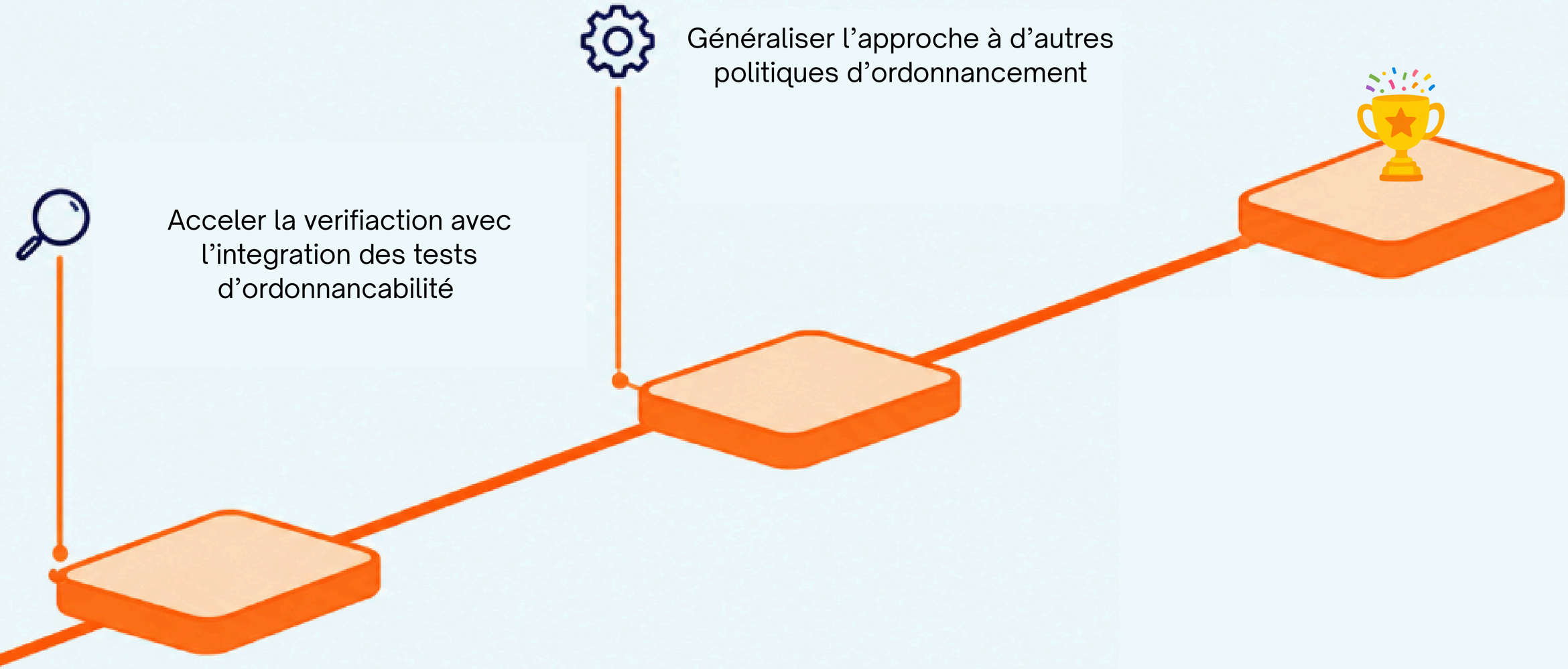
4. PERSPECTIVES

4. PERSPECTIVES



Acceler la verifiaction avec l’integration des tests d’ordonnancabilité

4. PERSPECTIVES



MERCI POUR VOTRE ATTENTION