

Exécution prédictible sur processeurs pluri-cœurs

Quentin Perret

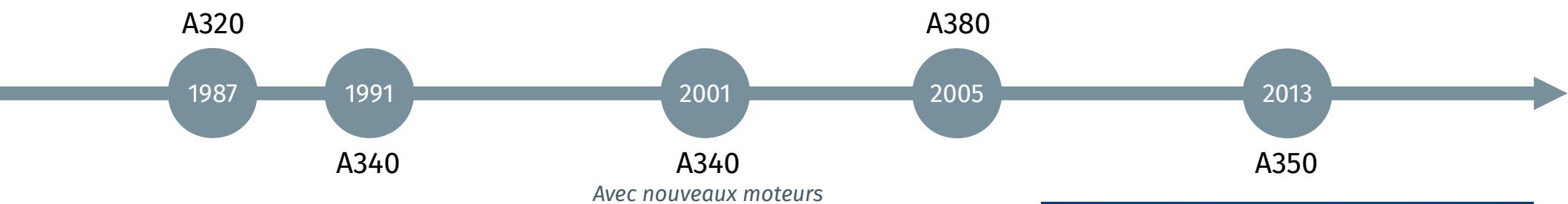
25 Avril 2017 - Toulouse, France

Composition du jury :

- M. Jean-Luc BECHENNEC, Chargé de recherche IRCCyN, Examinateur
- M. Emmanuel GROLLEAU, Professeur ISAE-ENSMA, Rapporteur
- Mme Claire MAIZA, Maître de Conférences INP Grenoble, Examinatrice
- M. Pascal MAURERE, Ingénieur Airbus Operations SAS, Encadrant industriel
- M. Eric NOULARD, Ingénieur de recherche ONERA, Invité
- Mme Claire PAGETTI, Maître de recherche ONERA, Directrice de thèse
- M. Laurent PAUTET, Professeur Télécom ParisTech, Rapporteur
- Mme Isabelle PUAUT, Professeur Université de Rennes I, Examinatrice
- M. Pascal SAINRAT, Professeur Université Toulouse III, Examinateur
- M. Benoit TRIQUET, Ingénieur Airbus Operations SAS, Invité



Contexte avionique



Contexte avionique



A320

1987

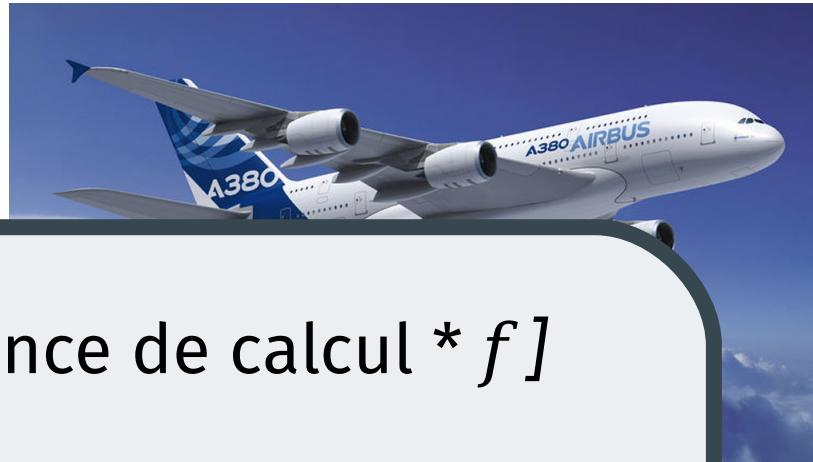
1994
A330

[Puissance de calcul * f]

avec

$$2 \leq f \leq 3$$

à chaque génération



2013
A350



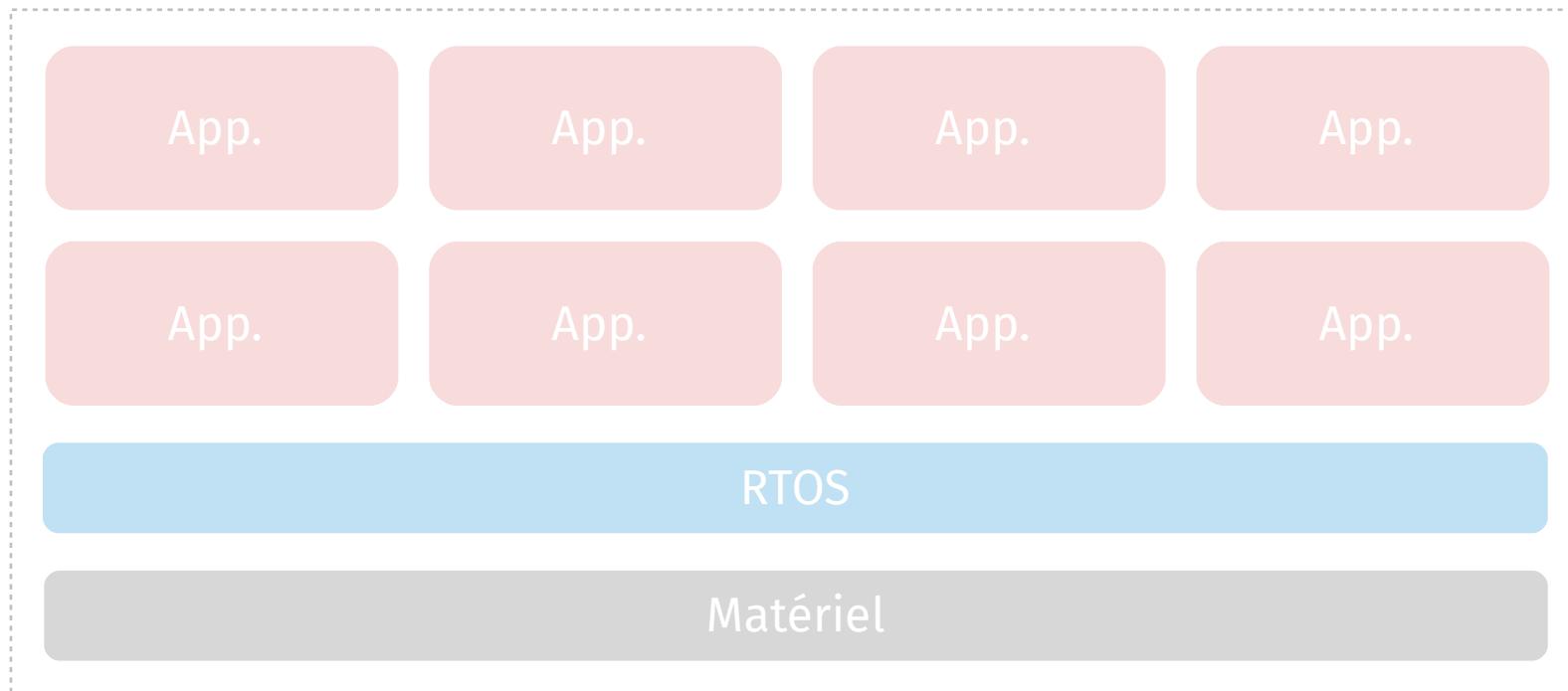
Contexte industriel



Approche fédérée

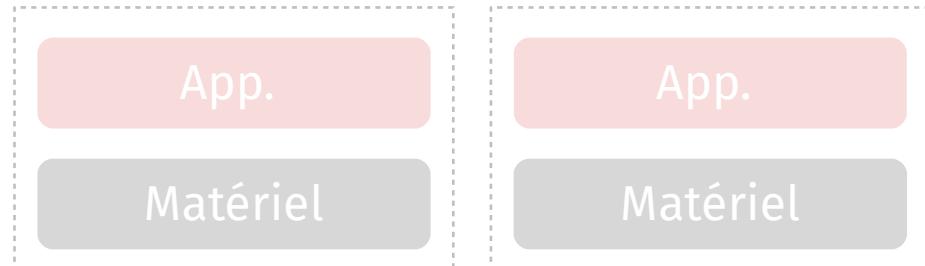


Avionique Modulaire Intégrée (IMA)

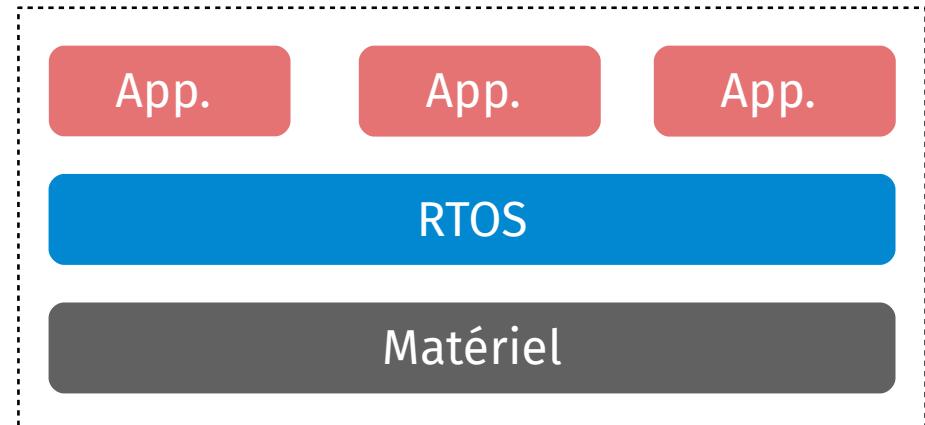


Futur

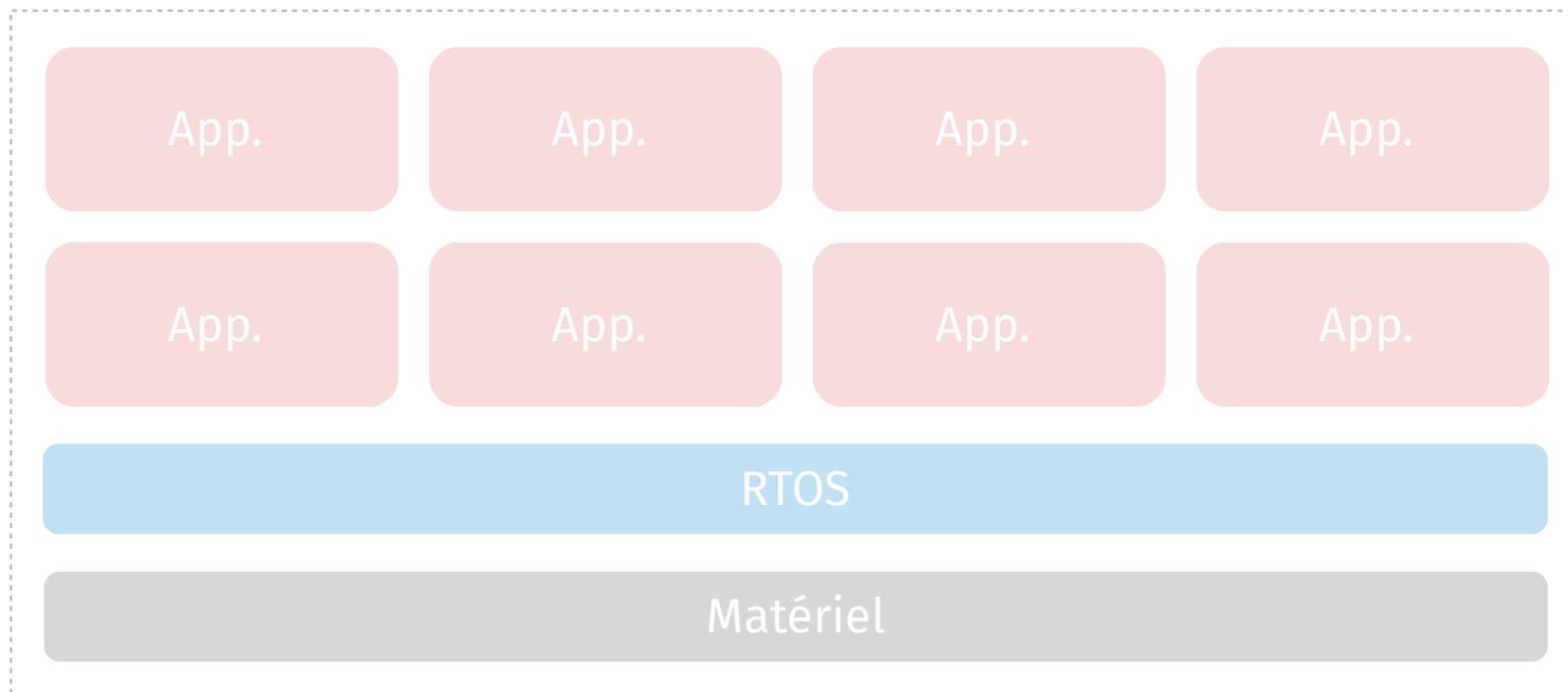
Contexte industriel



Approche fédérée

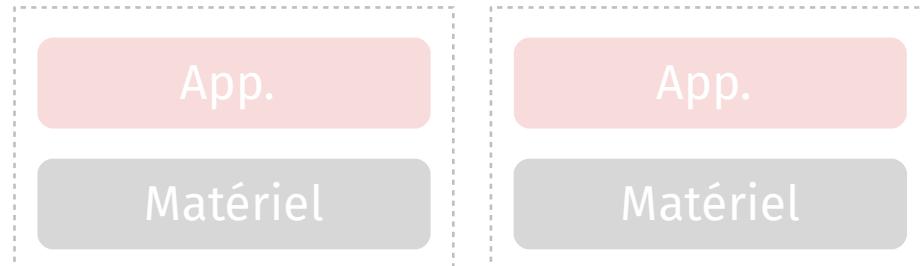


Avionique Modulaire Intégrée (IMA)

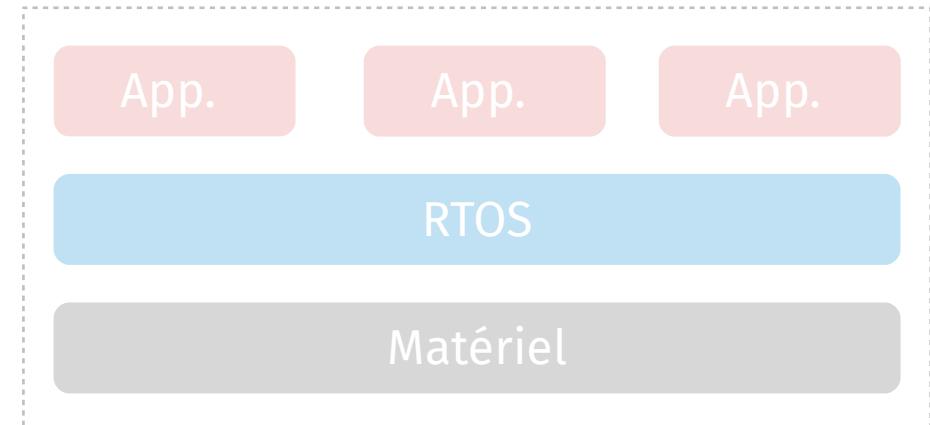


Futur

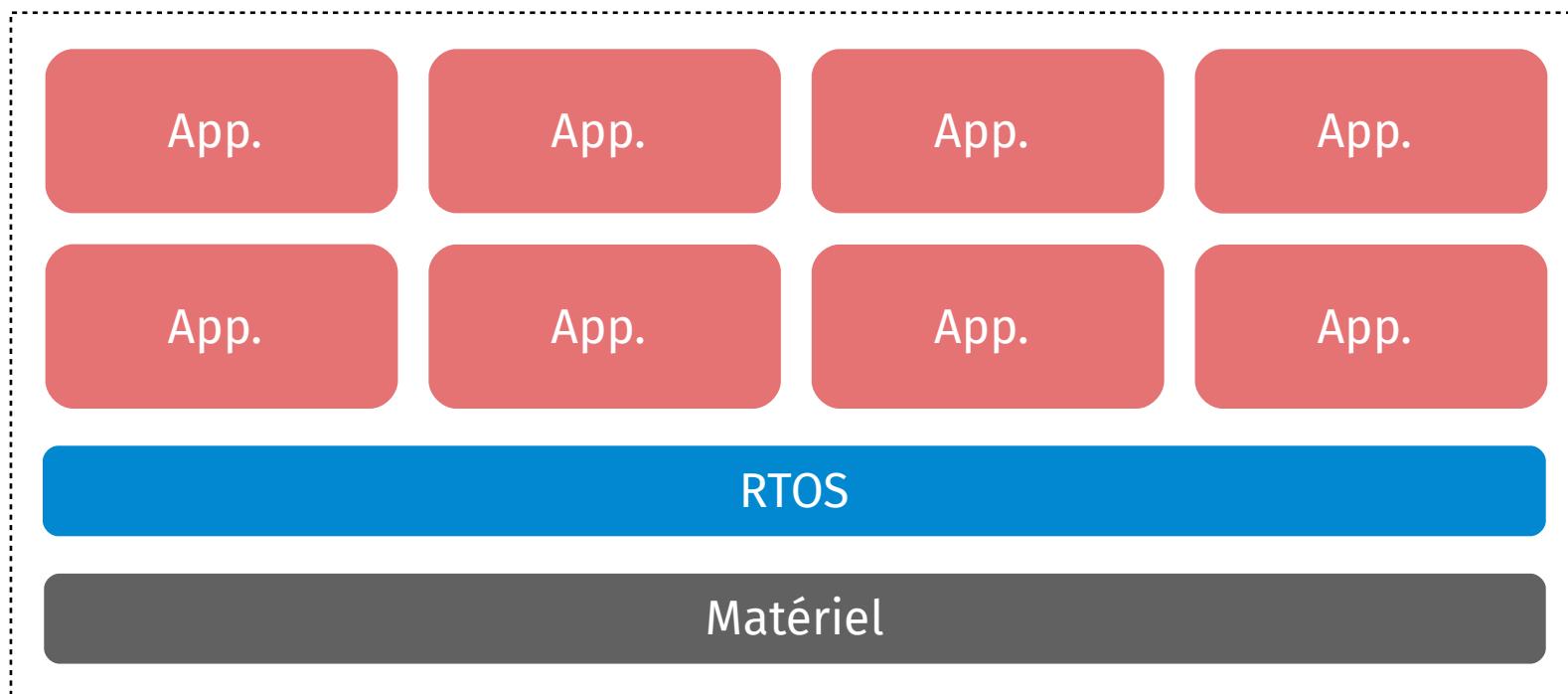
Contexte industriel



Approche fédérée

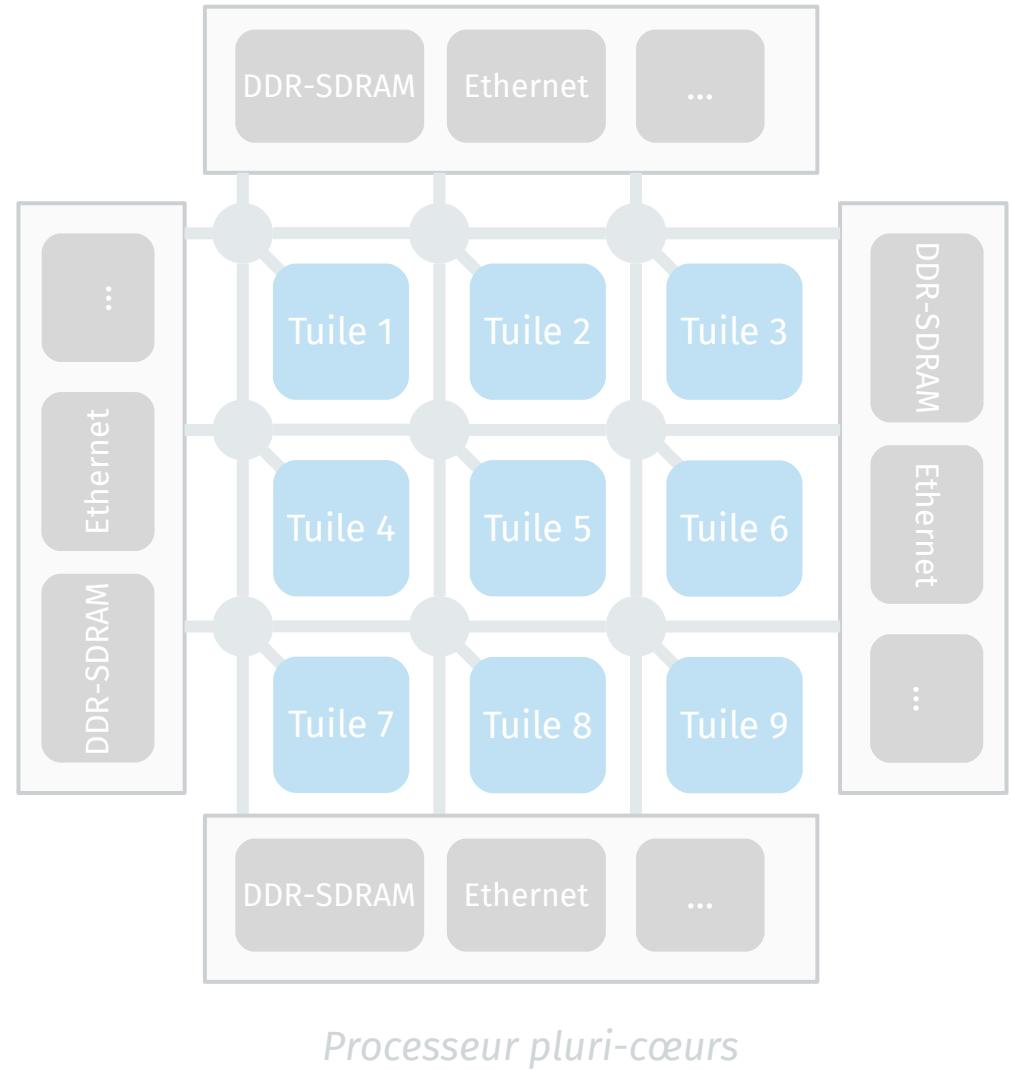
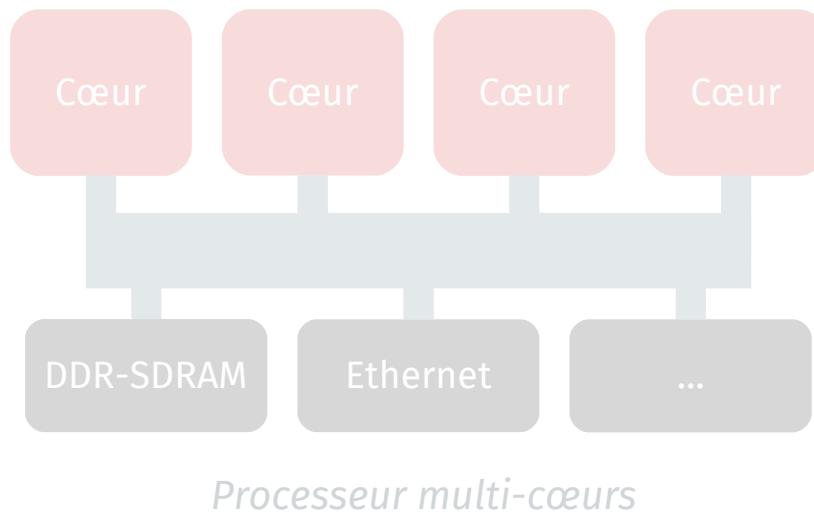
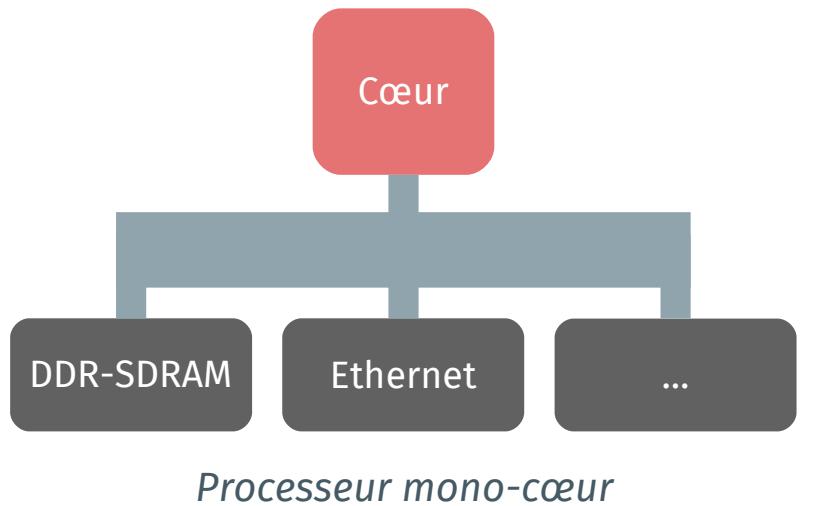


Avionique Modulaire Intégrée (IMA)

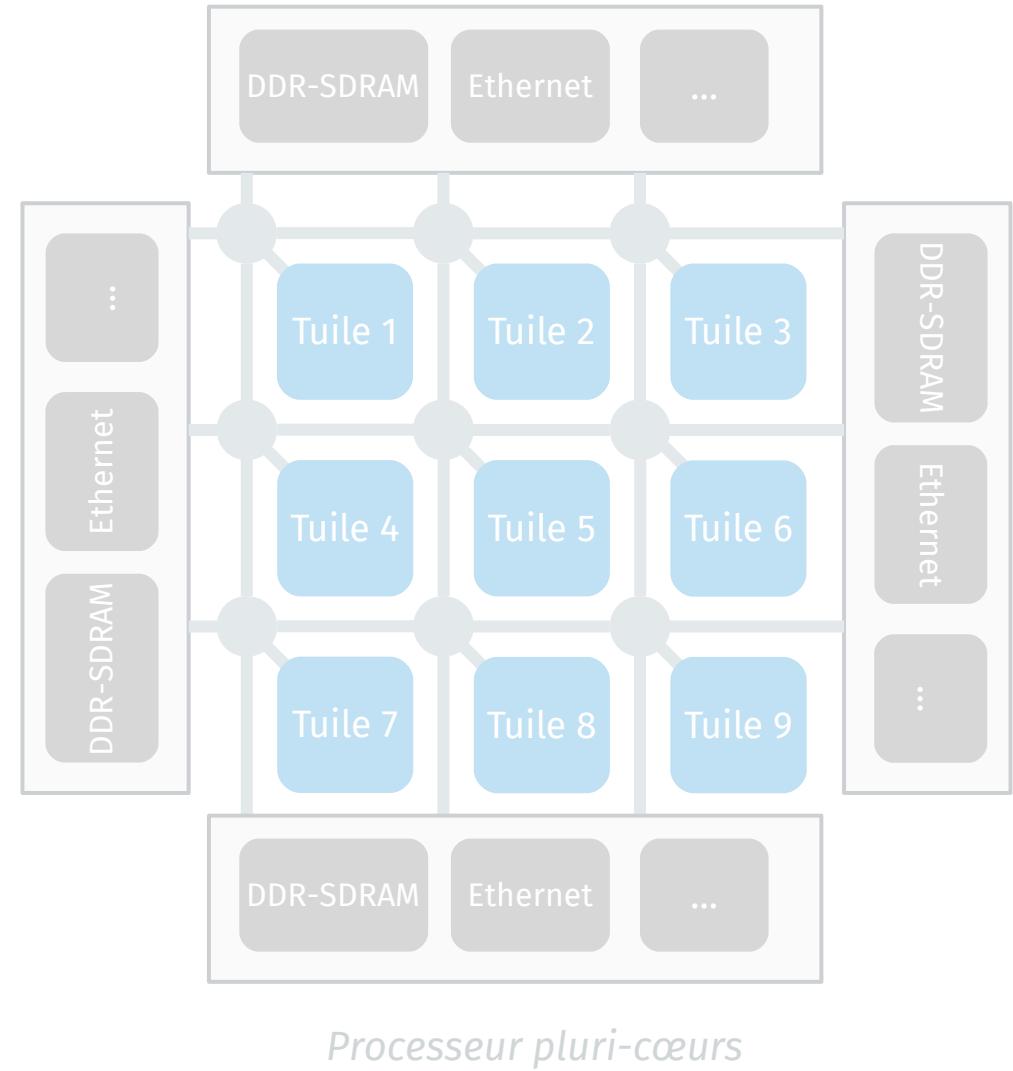
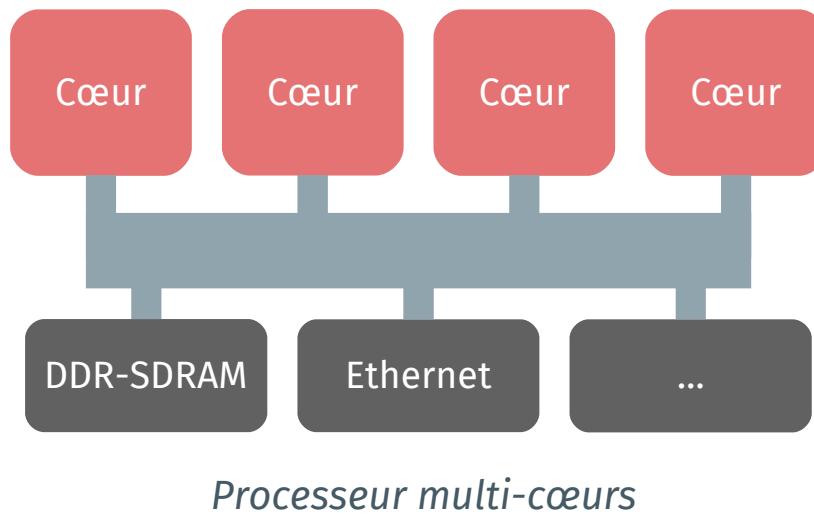
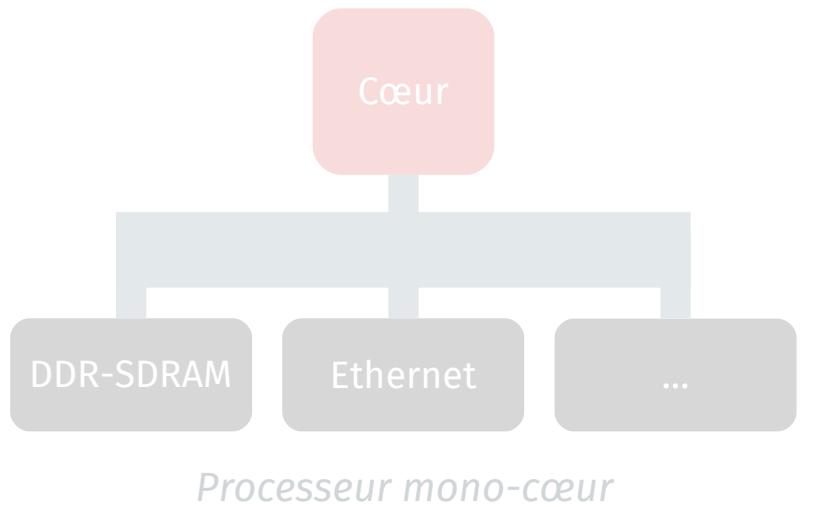


Futur

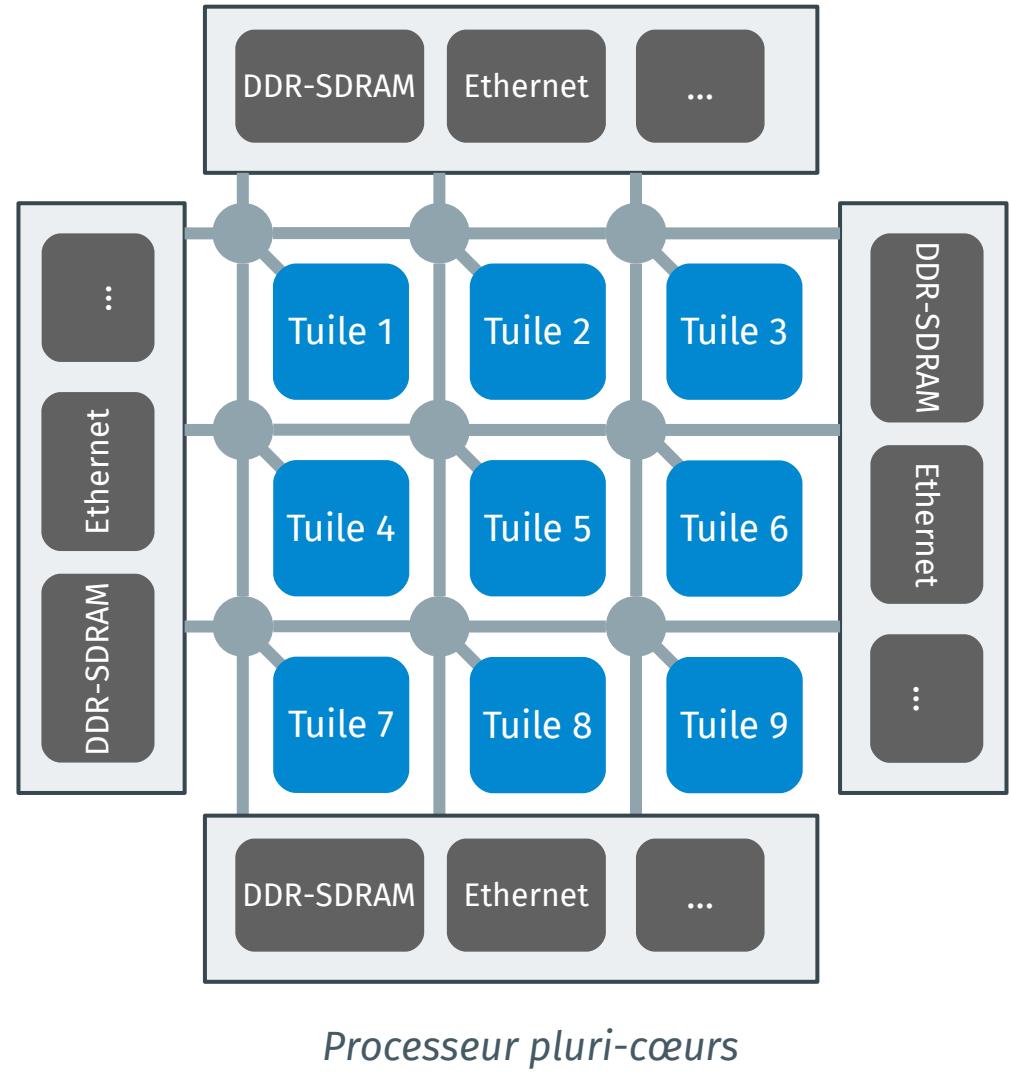
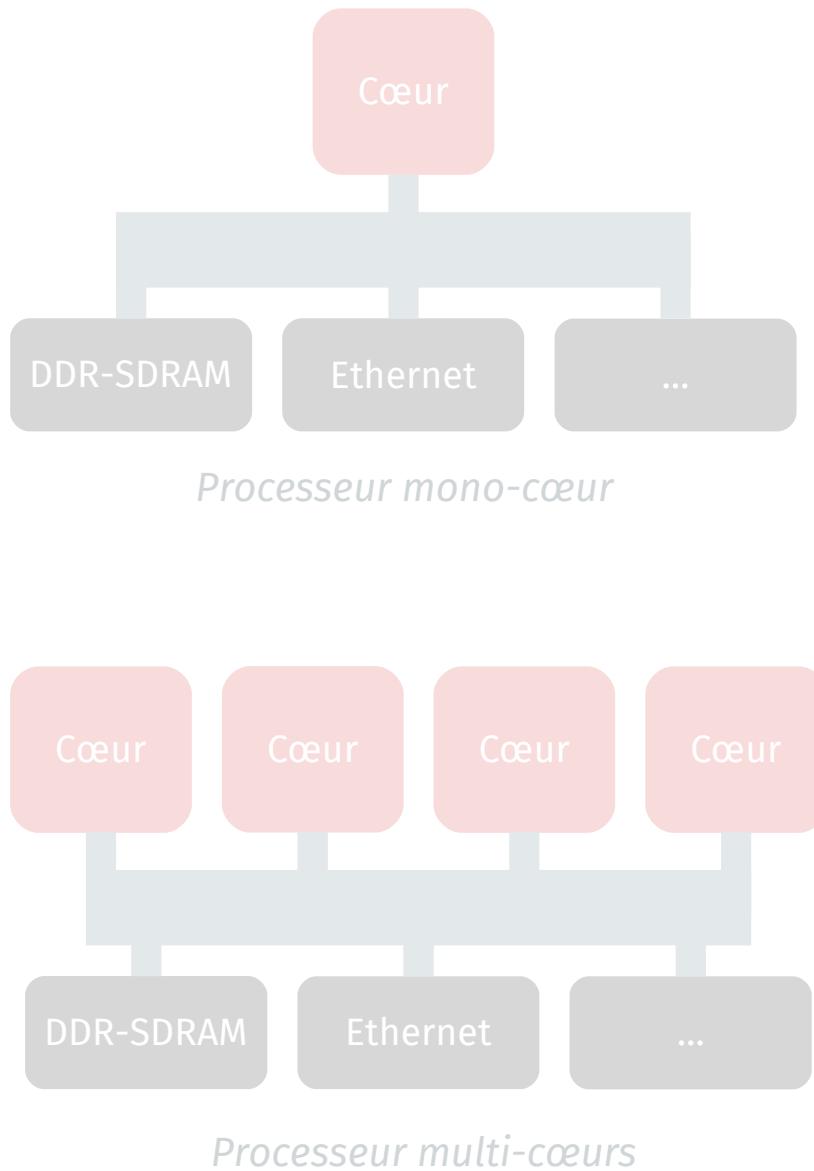
Contexte technologique



Contexte technologique



Contexte technologique



Problématiques & contributions

Problématique 1 :

Comment **partager** un processeur pluri-cœurs ?

Problématiques & contributions

Problématique 1 :

Comment **partager** un processeur pluri-cœurs ?

- Attentes : conception, développement, test et certification des systèmes en ***indépendance***

Problématiques & contributions

Problématique 1 :

Comment **partager** un processeur pluri-cœurs ?

- Attentes : conception, développement, test et certification des systèmes en **indépendance**
- Contribution : Définition d'un **modèle d'exécution** pour assurer une **isolation temporelle** et implémentation par un **hyperviseur**

Problématiques & contributions

Problématique 1 :

Comment **partager** un processeur pluri-cœurs ?

- Attentes : conception, développement, test et certification des systèmes en **indépendance**
- Contribution : Définition d'un **modèle d'exécution** pour assurer une **isolation temporelle** et implémentation par un **hyperviseur**

Problématique 2 :

Comment **paralléliser** nos applications ?

Problématiques & contributions

Problématique 1 :

Comment **partager** un processeur pluri-cœurs ?

- Attentes : conception, développement, test et certification des systèmes en **indépendance**
- Contribution : Définition d'un **modèle d'exécution** pour assurer une **isolation temporelle** et implémentation par un **hyperviseur**

Problématique 2 :

Comment **paralléliser** nos applications ?

- Attentes : Parallélisation **efficace** et **automatique** d'applications de **grandes tailles**

Problématiques & contributions

Problématique 1 :

Comment **partager** un processeur pluri-cœurs ?

- Attentes : conception, développement, test et certification des systèmes en **indépendance**
- Contribution : Définition d'un **modèle d'exécution** pour assurer une **isolation temporelle** et implémentation par un **hyperviseur**

Problématique 2 :

Comment **paralléliser** nos applications ?

- Attentes : Parallélisation **efficace** et **automatique** d'applications de **grandes tailles**
- Contribution : **Formalisation** du problème permettant une **Résolution rapide** à l'aide d'un solveur de contraintes

Plan

1. Kalray MPPA-256
2. Première contribution : Isolation temporelle
3. Deuxième contribution : Placement automatique
4. Conclusion

Plan

1. Kalray MPPA-256

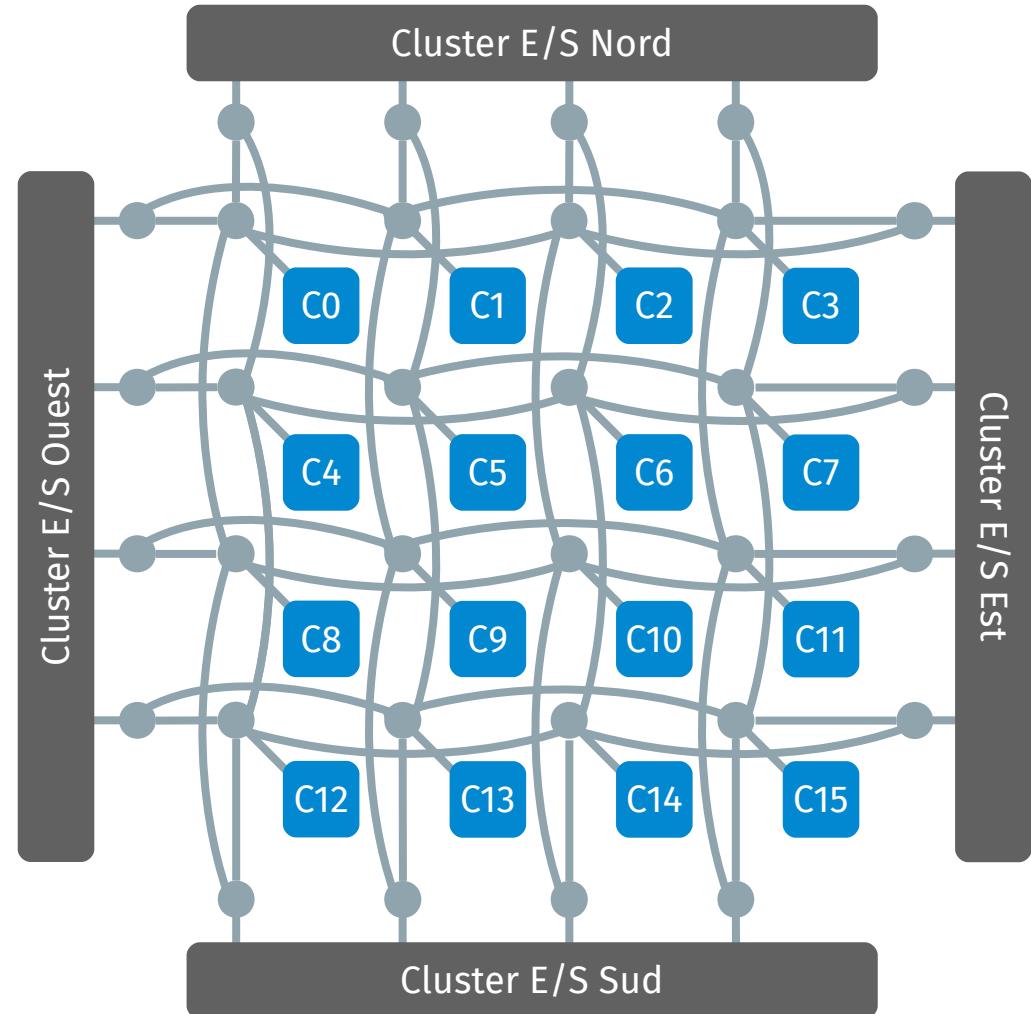
2. Première contribution : Isolation temporelle

3. Deuxième contribution : Placement automatique

4. Conclusion

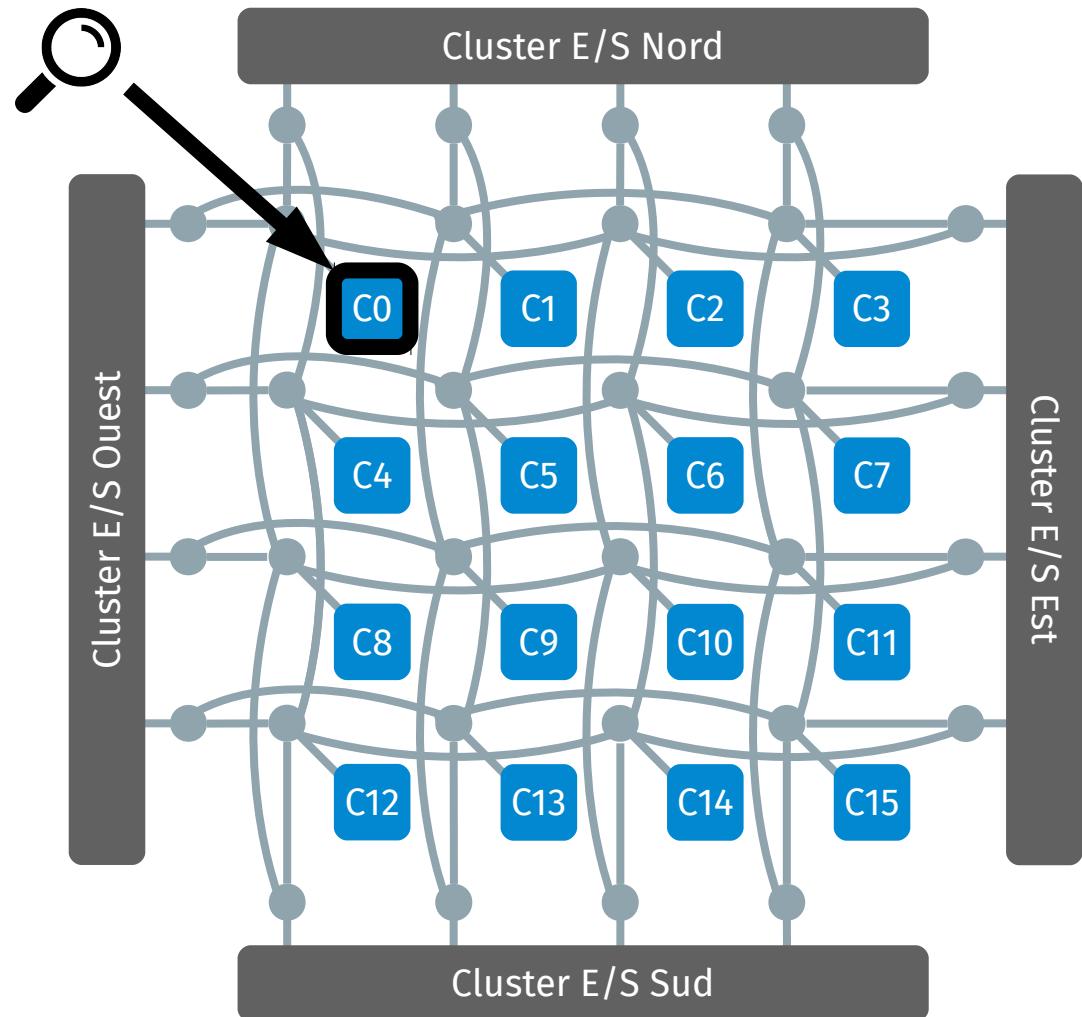
Vue d'ensemble

- 256 coeurs
- Très programmable
- Info. disponibles
- Bonnes propriétés temporelles
- Conso. < 20W possible



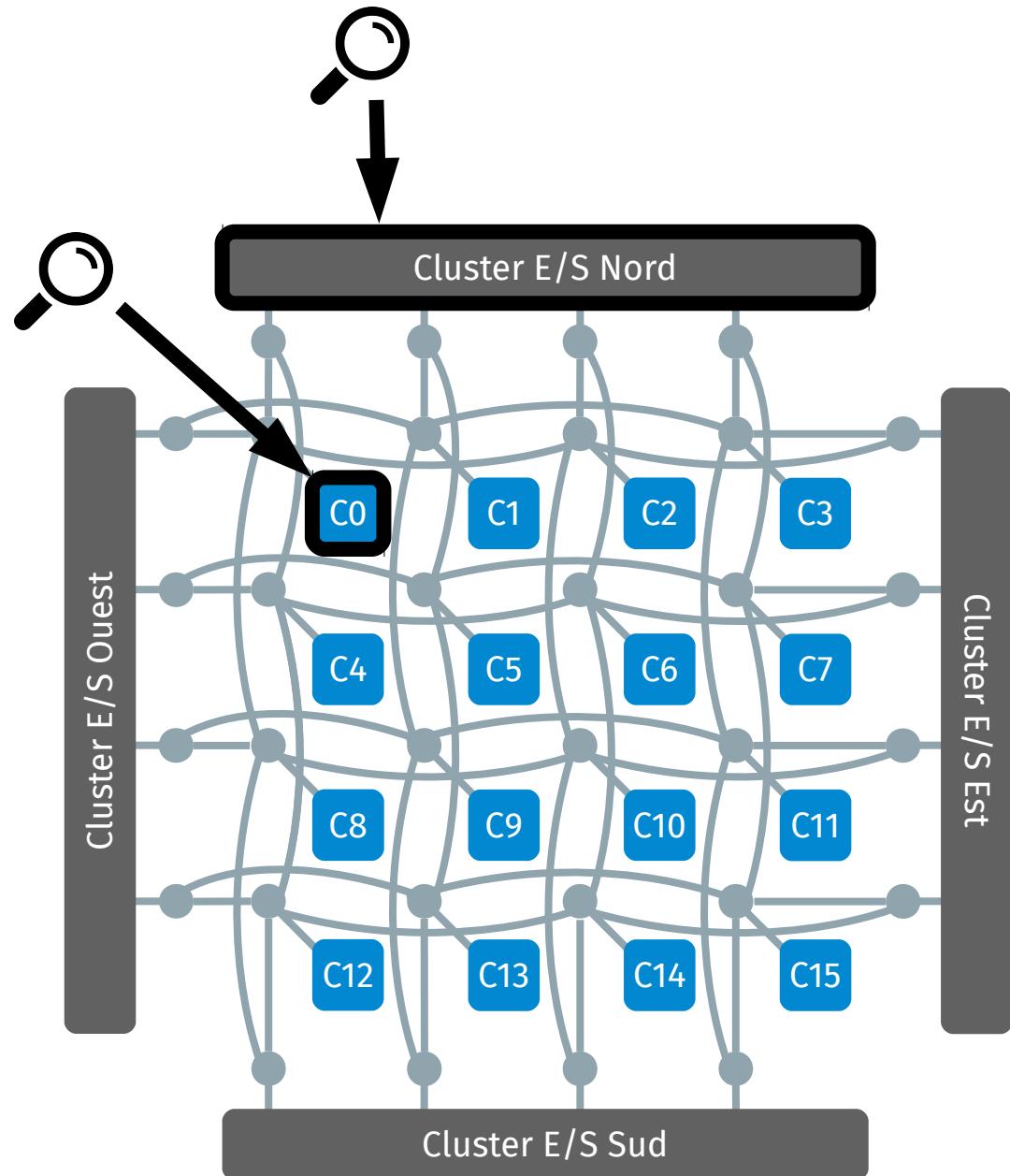
Vue d'ensemble

- 256 coeurs
- Très programmable
- Info. disponibles
- Bonnes propriétés temporelles
- Conso. < 20W possible

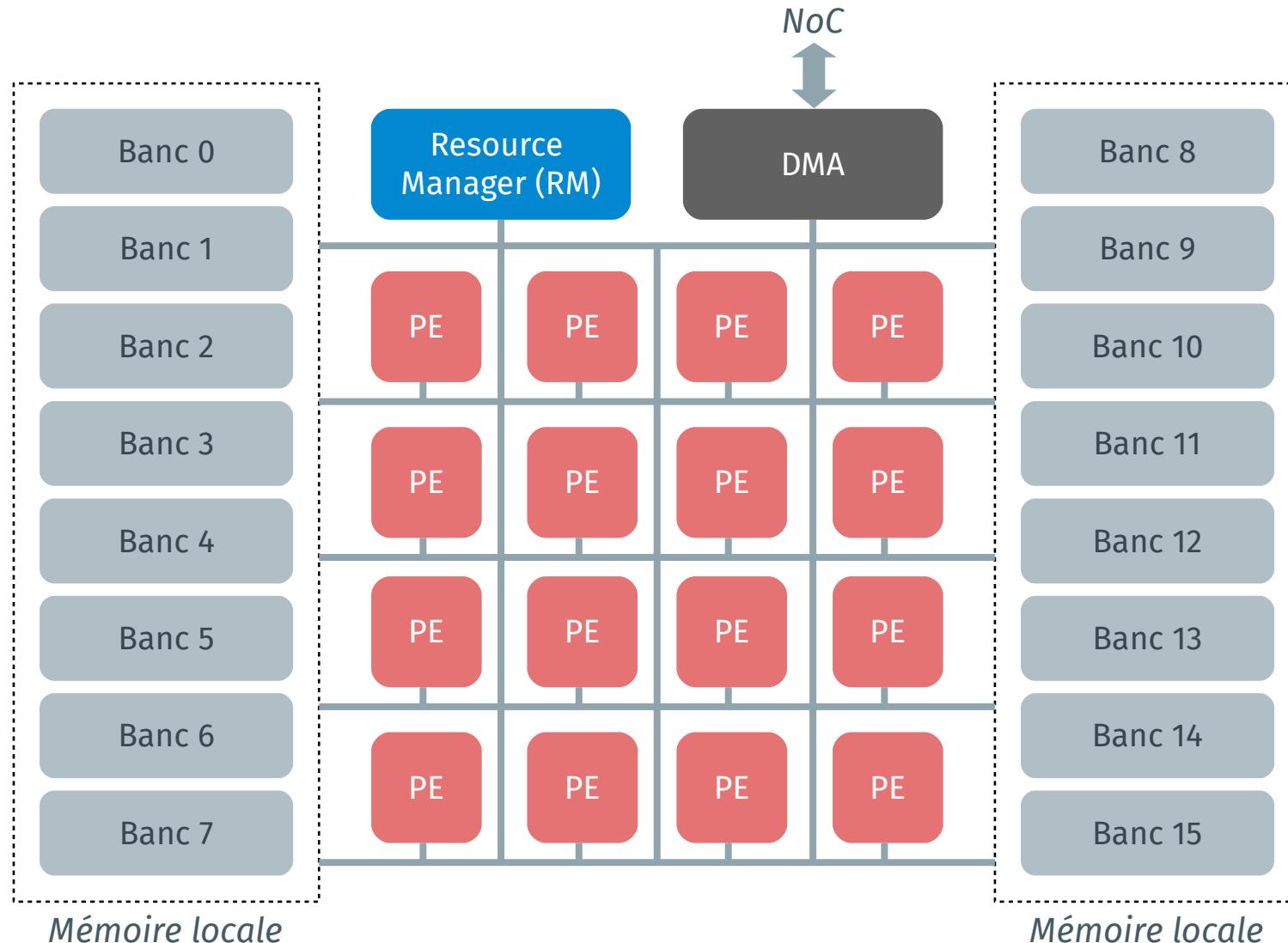


Vue d'ensemble

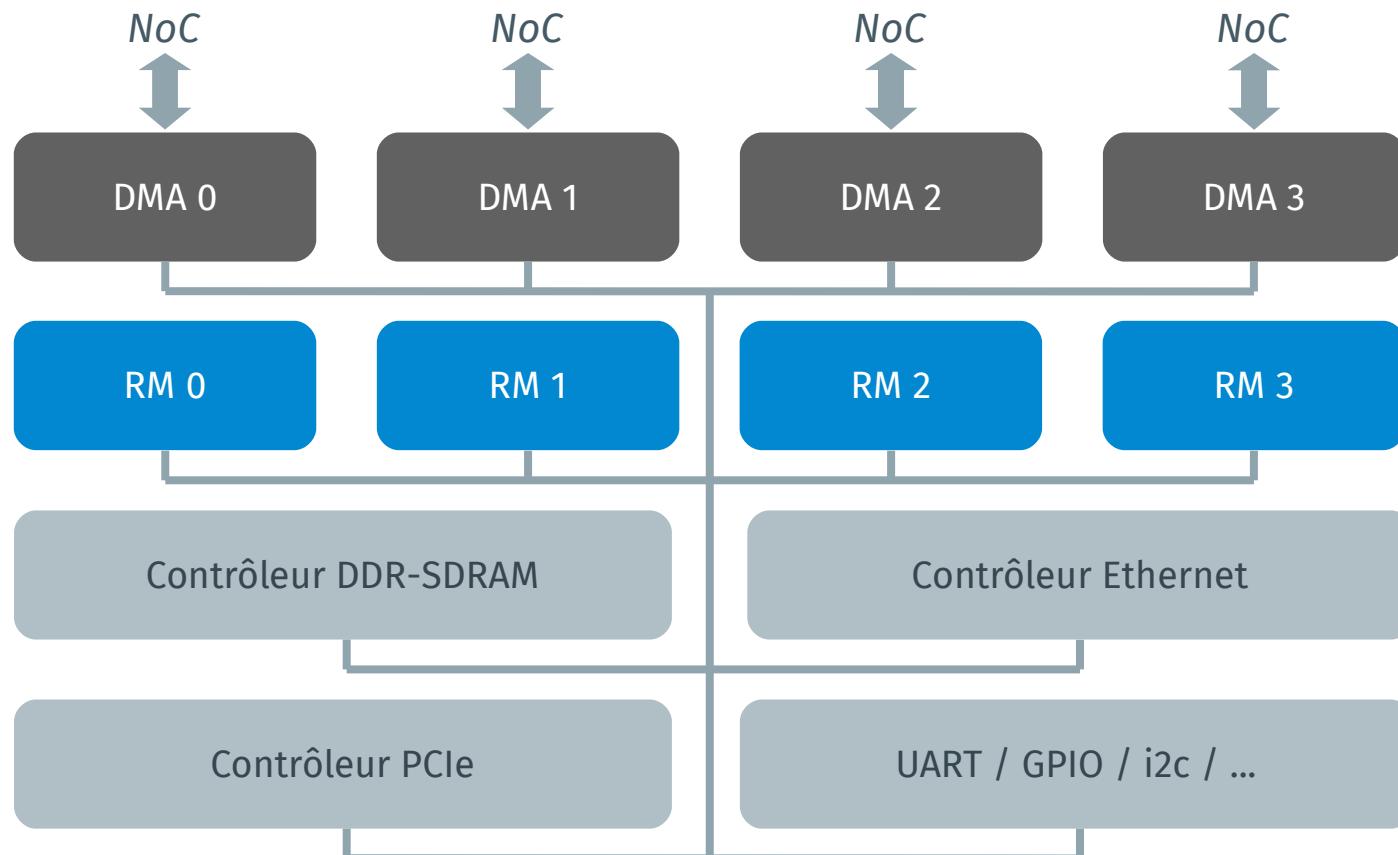
- 256 coeurs
- Très programmable
- Info. disponibles
- Bonnes propriétés temporelles
- Conso. < 20W possible



Clusters de calcul



Clusters E/S



Plan

1. Kalray MPPA-256

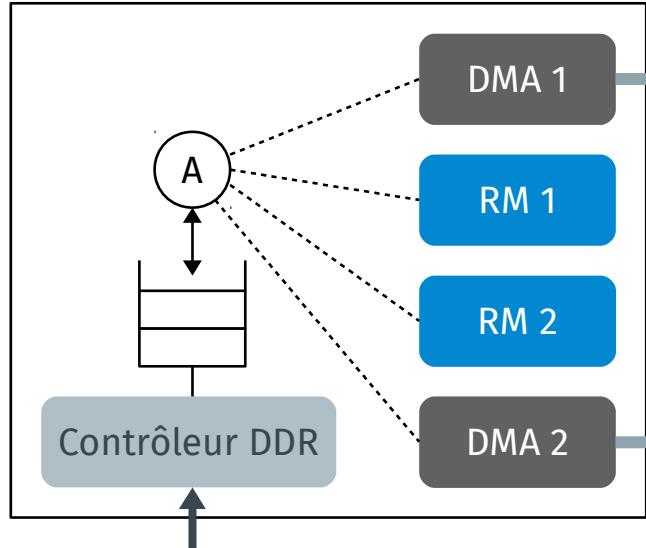
2. Première contribution : Isolation temporelle

3. Deuxième contribution : Placement automatique

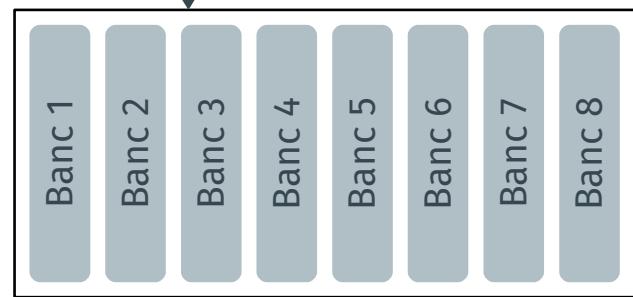
4. Conclusion

Accès à la DDR-SDRAM

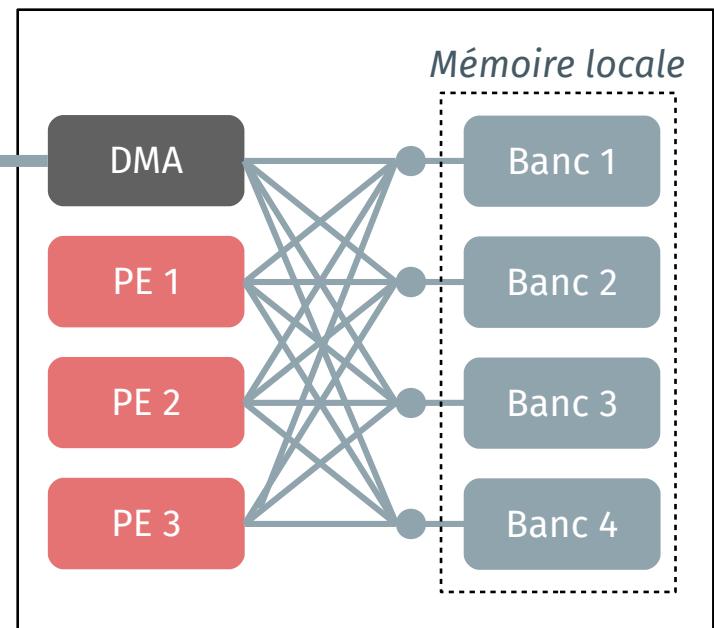
Cluster E/S



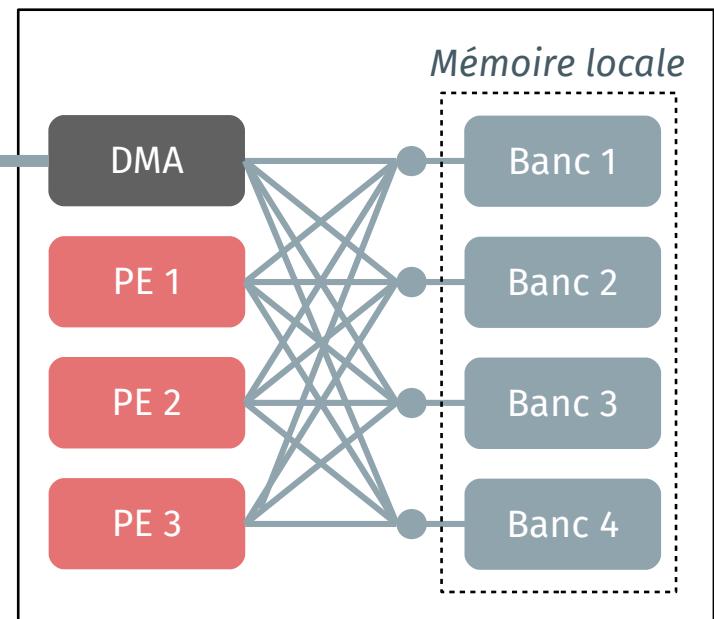
DDR-SDRAM



Cluster de calcul 1

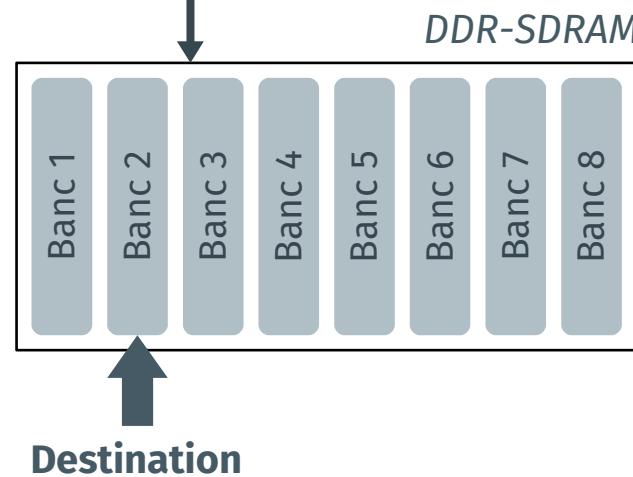
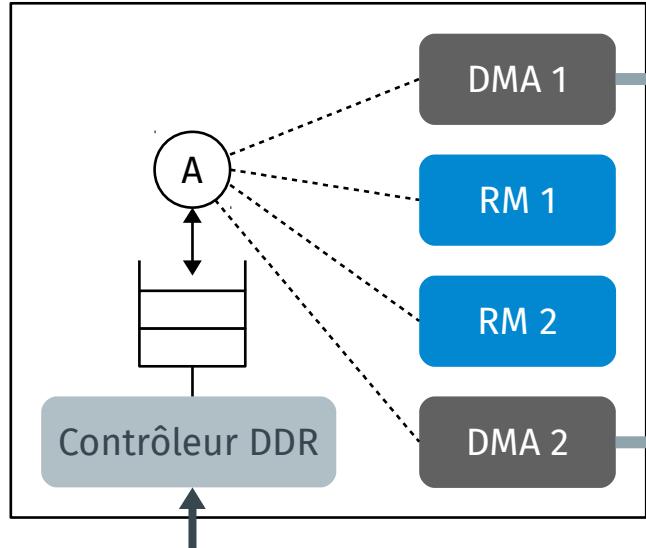


Cluster de calcul 2

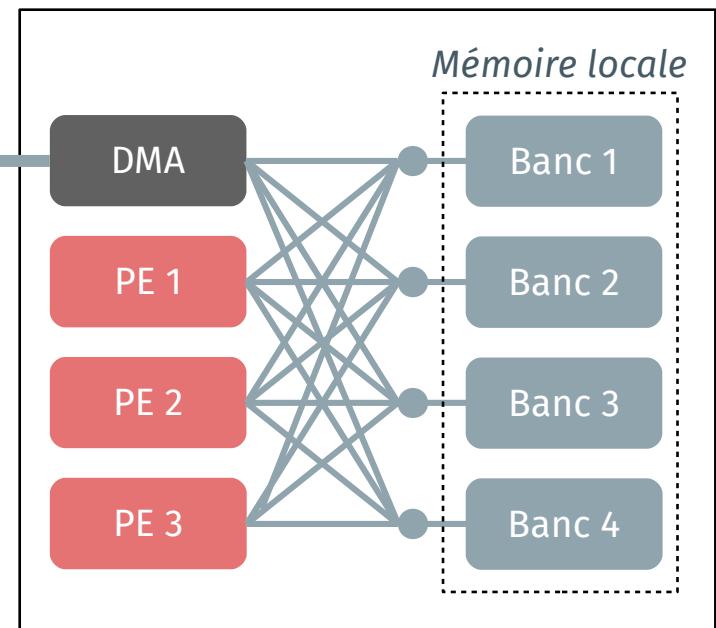


Accès à la DDR-SDRAM

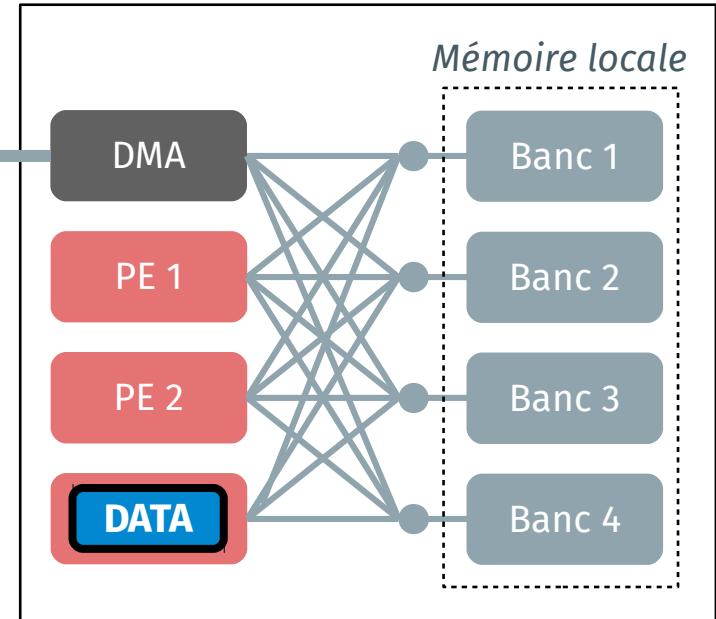
Cluster E/S



Cluster de calcul 1

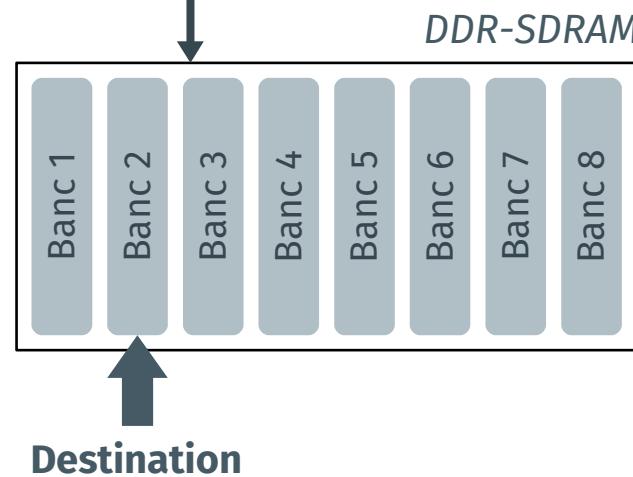
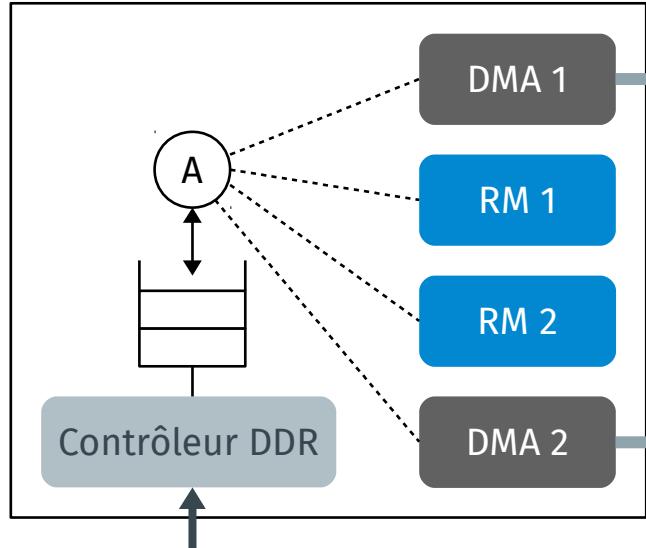


Cluster de calcul 2

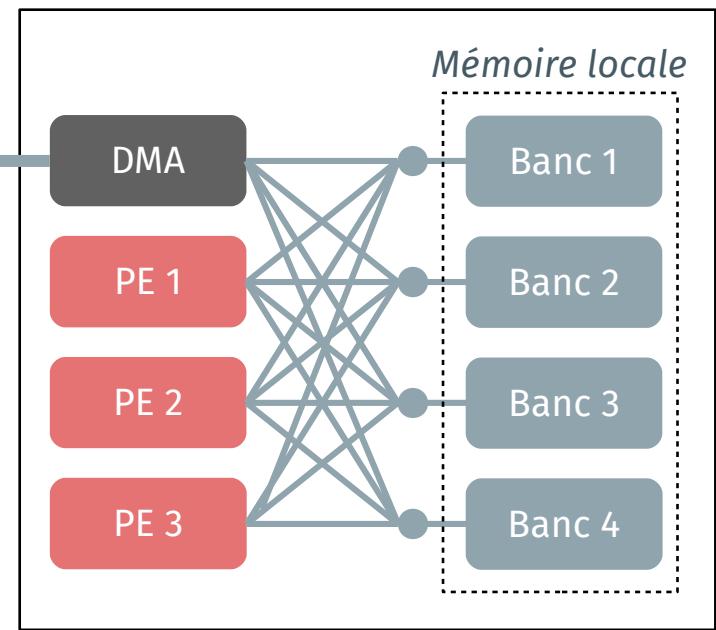


Accès à la DDR-SDRAM

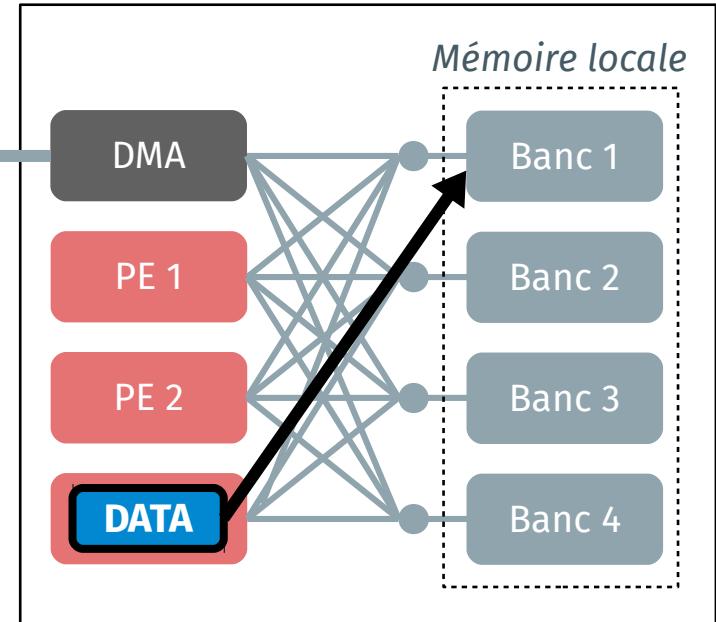
Cluster E/S



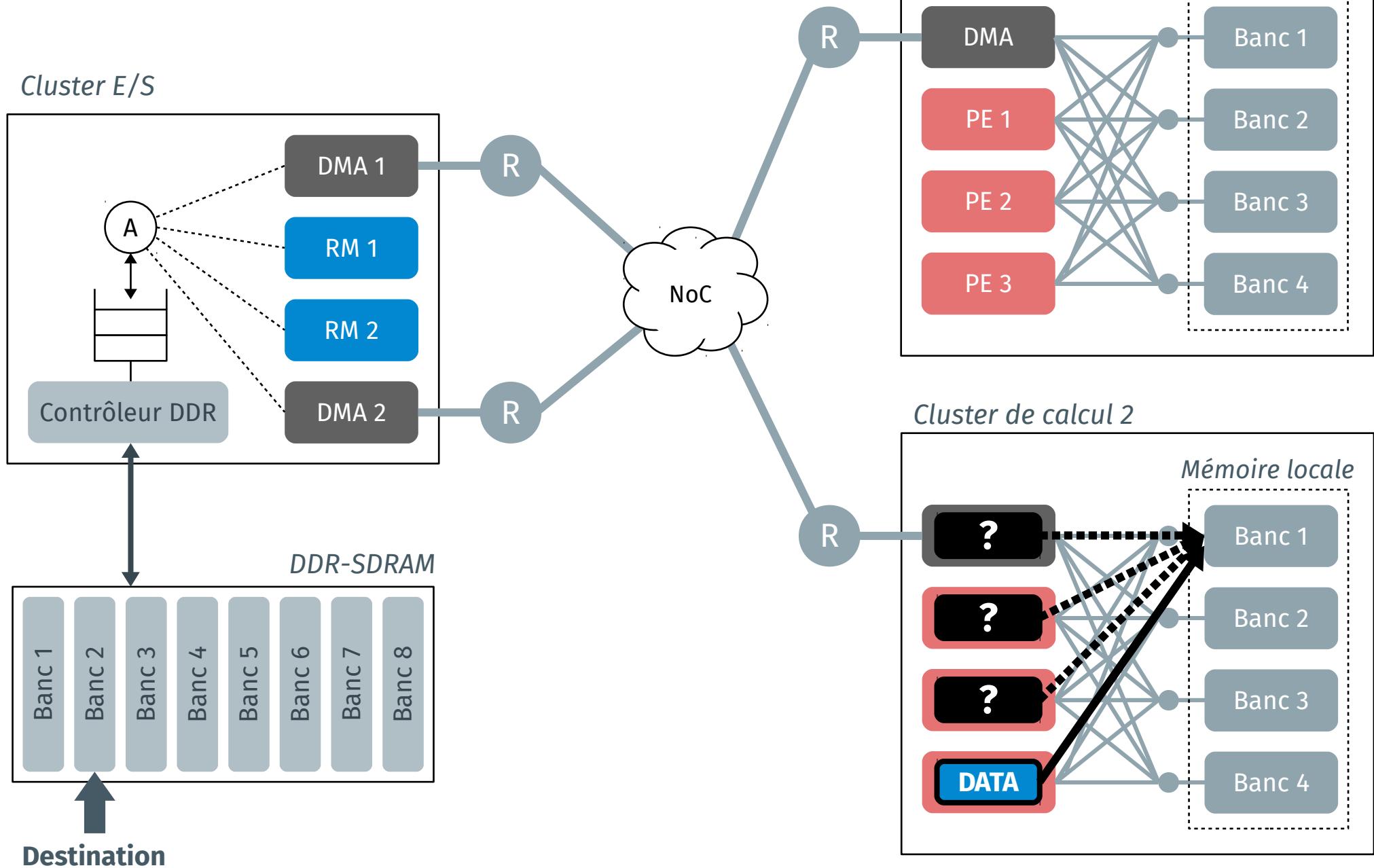
Cluster de calcul 1



Cluster de calcul 2

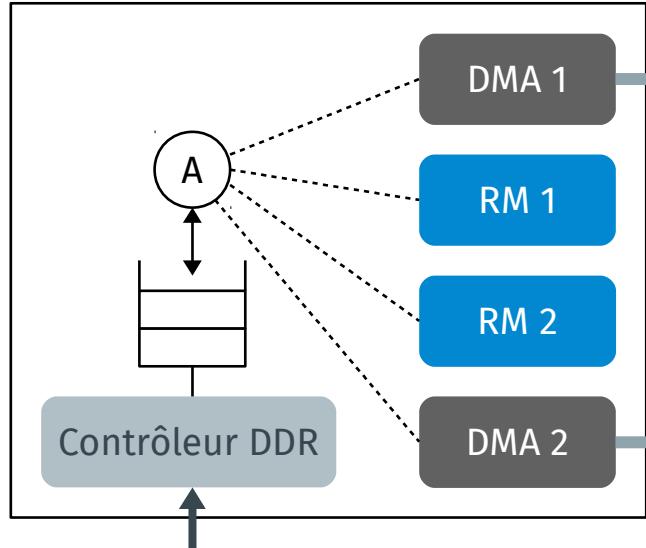


Accès à la DDR-SDRAM

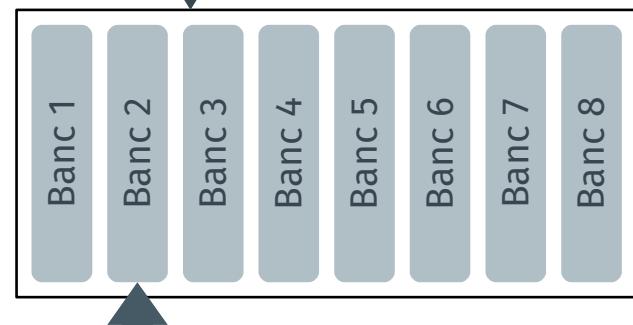


Accès à la DDR-SDRAM

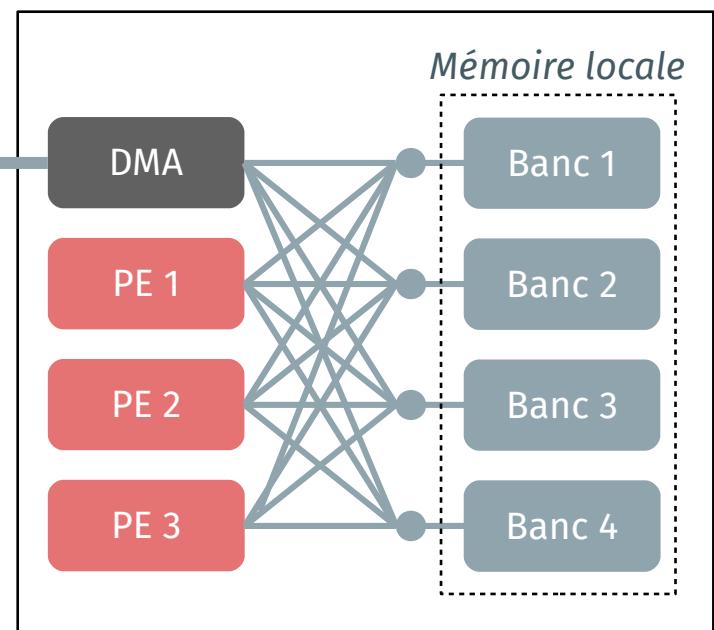
Cluster E/S



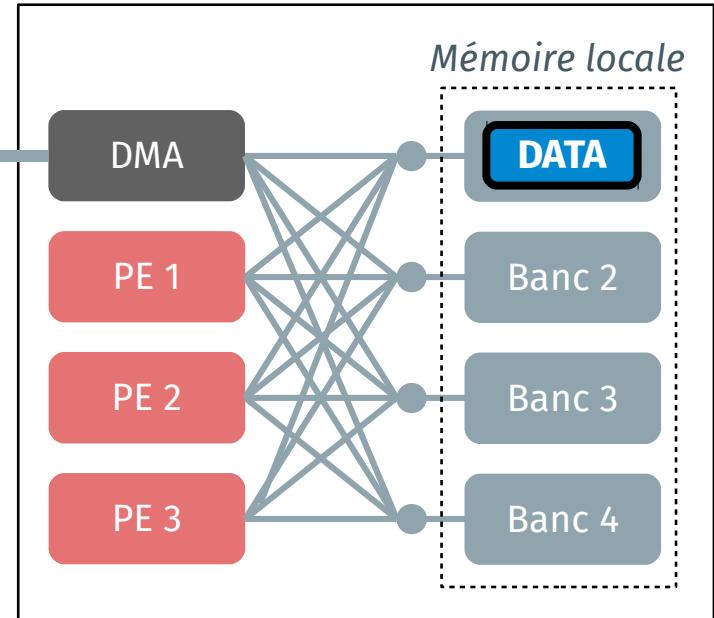
DDR-SDRAM



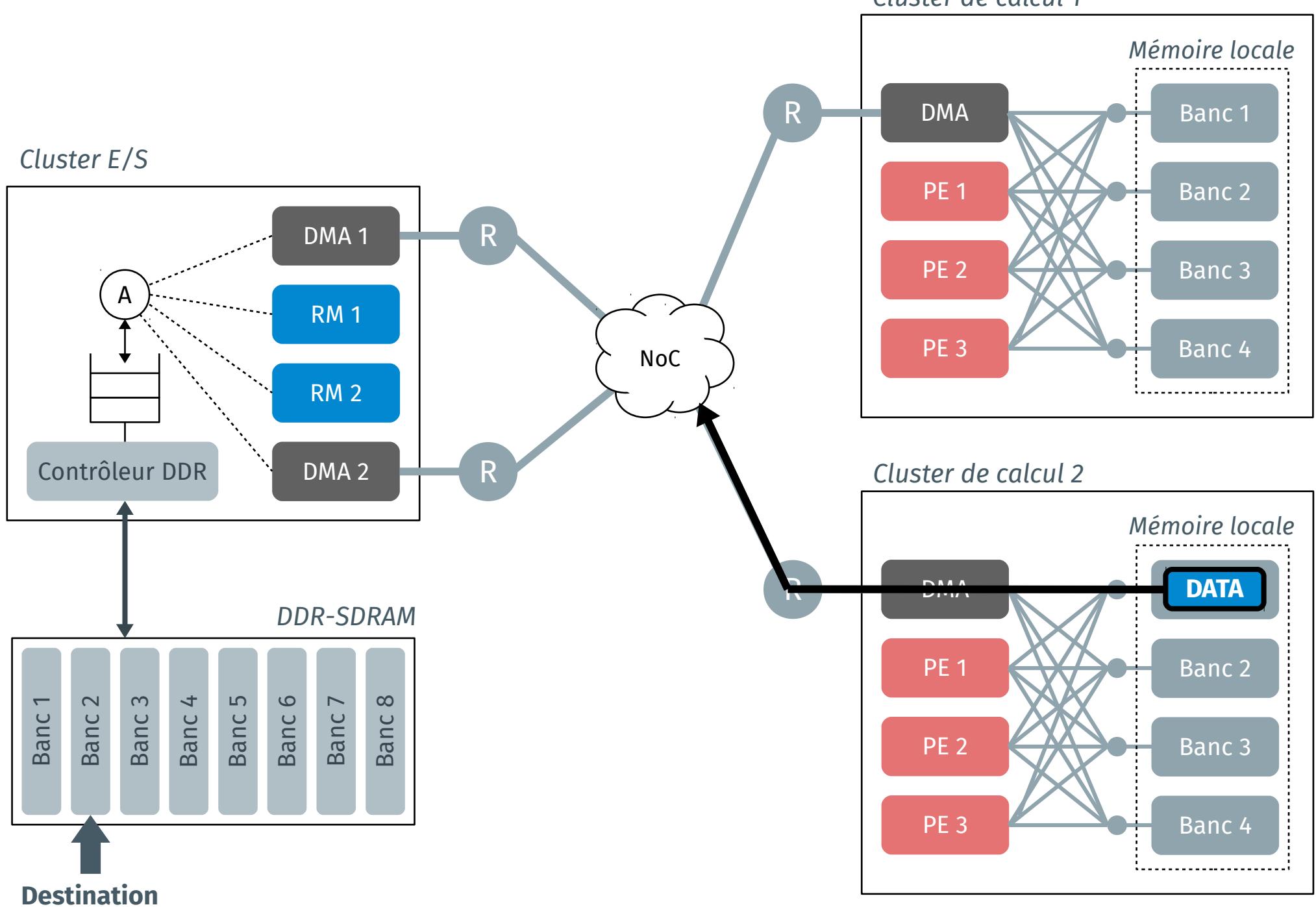
Cluster de calcul 1



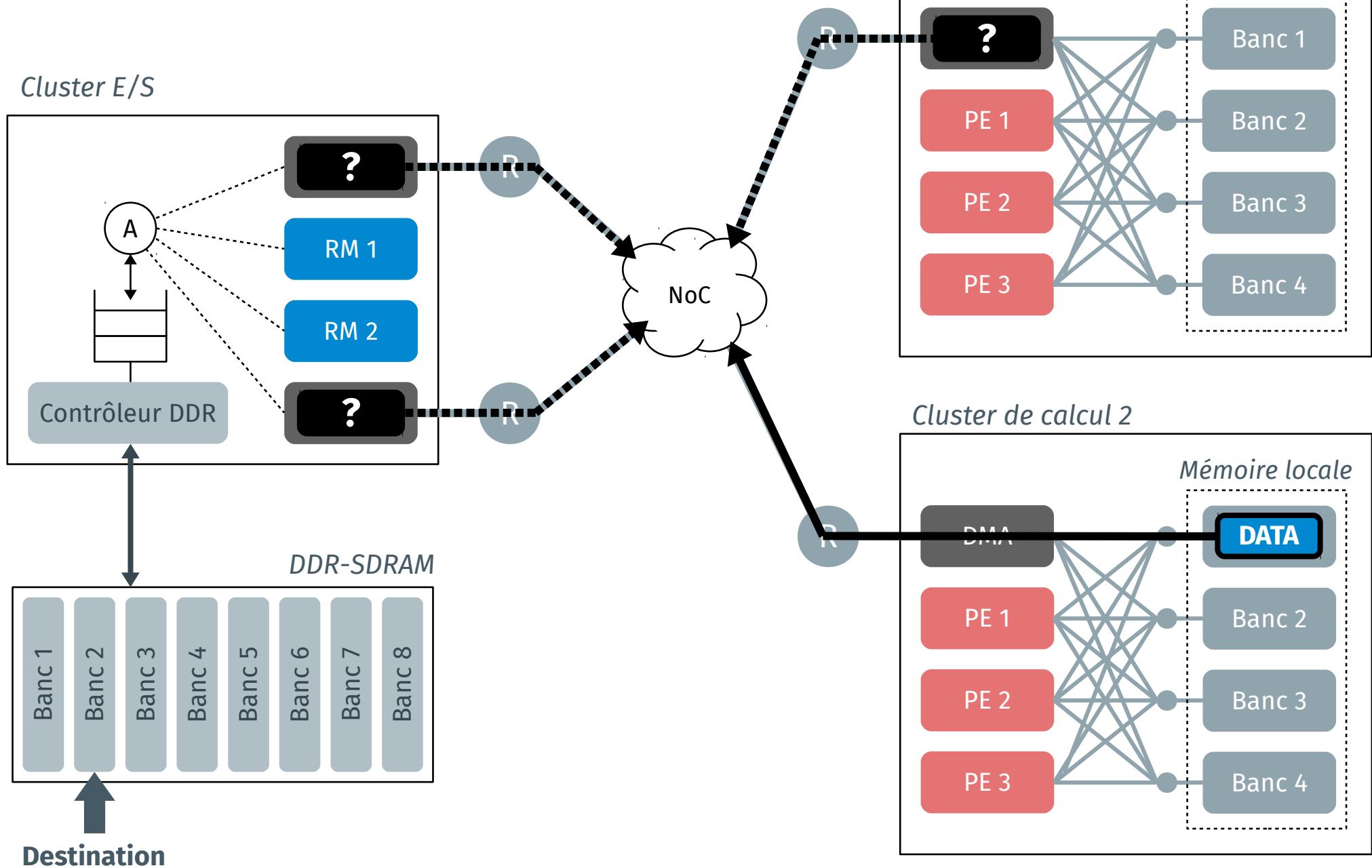
Cluster de calcul 2



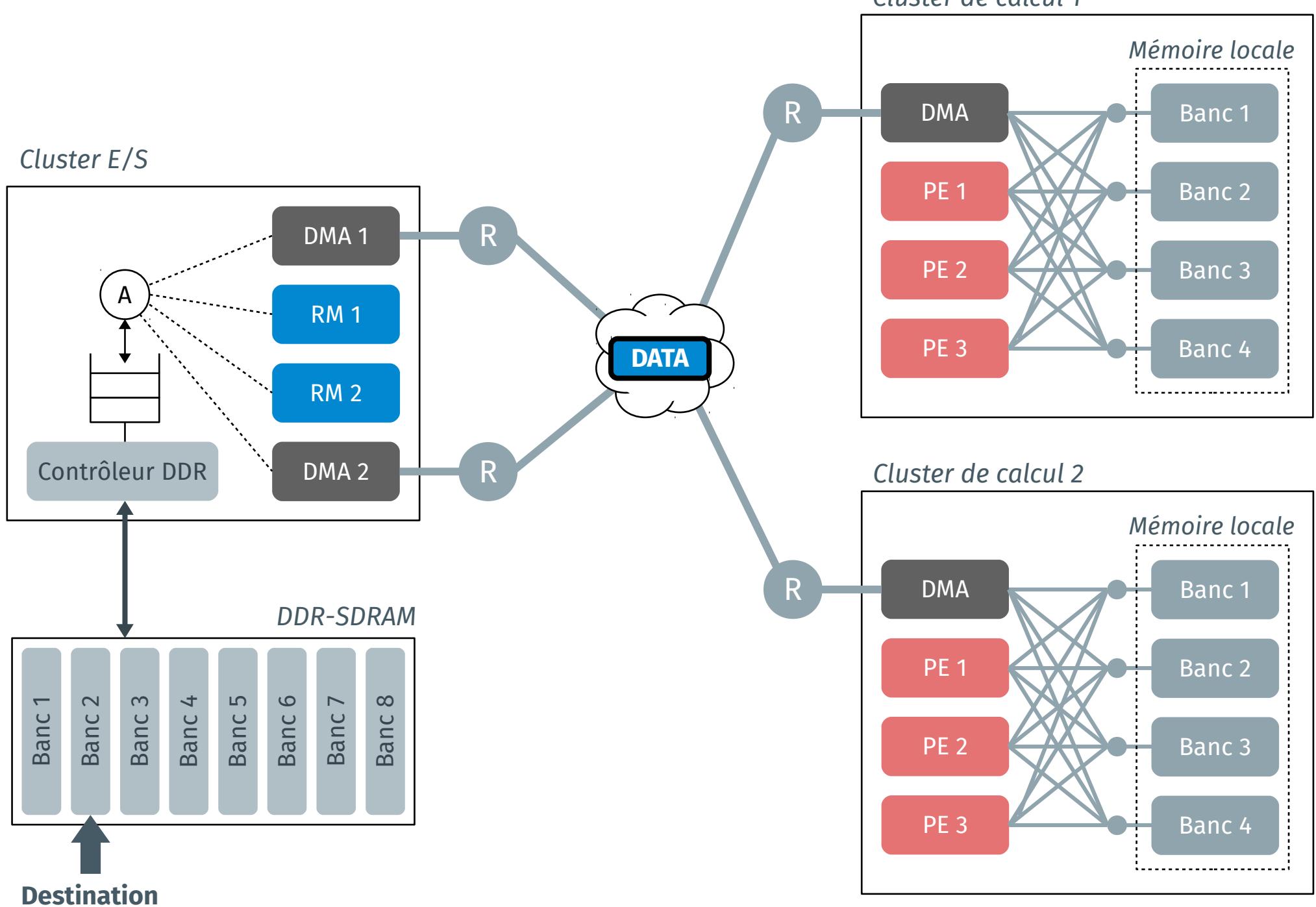
Accès à la DDR-SDRAM



Accès à la DDR-SDRAM

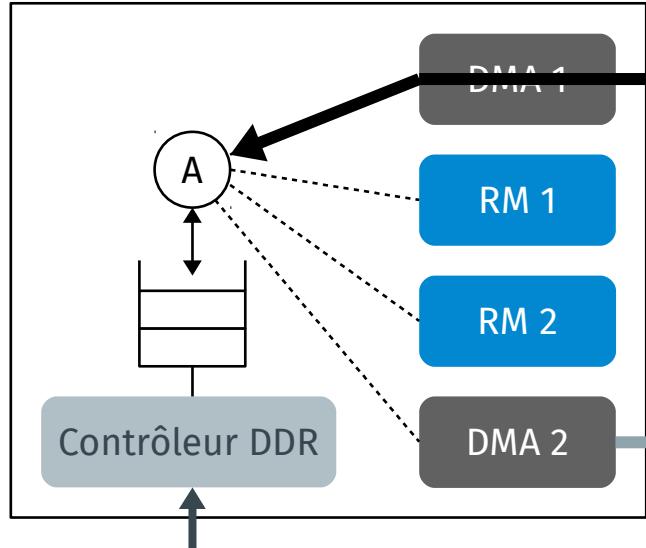


Accès à la DDR-SDRAM



Accès à la DDR-SDRAM

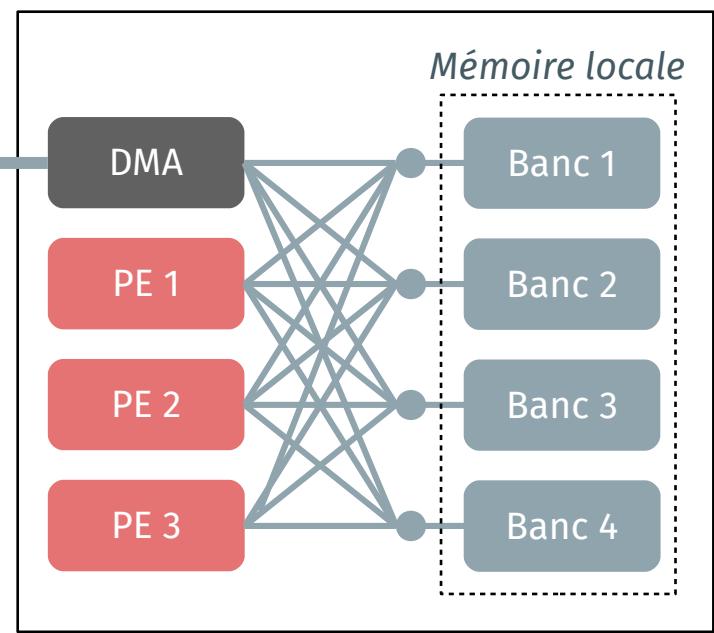
Cluster E/S



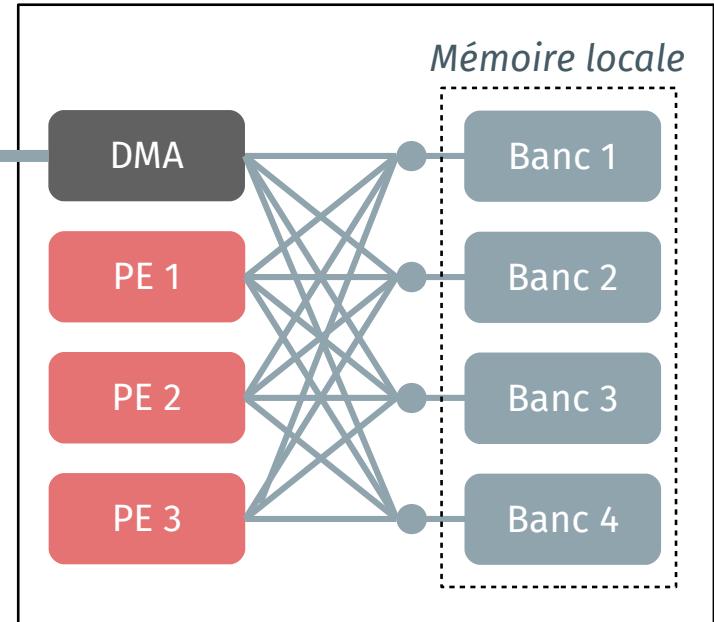
DDR-SDRAM

Destination

Cluster de calcul 1

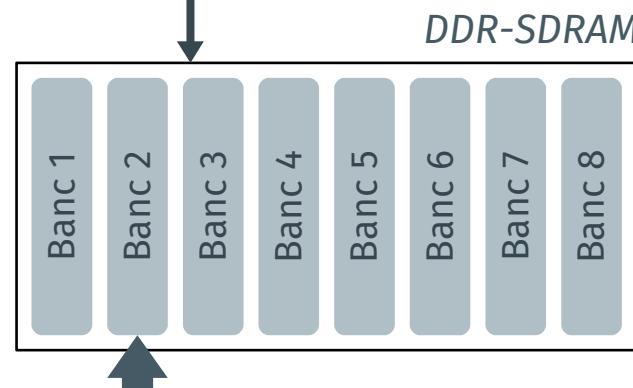
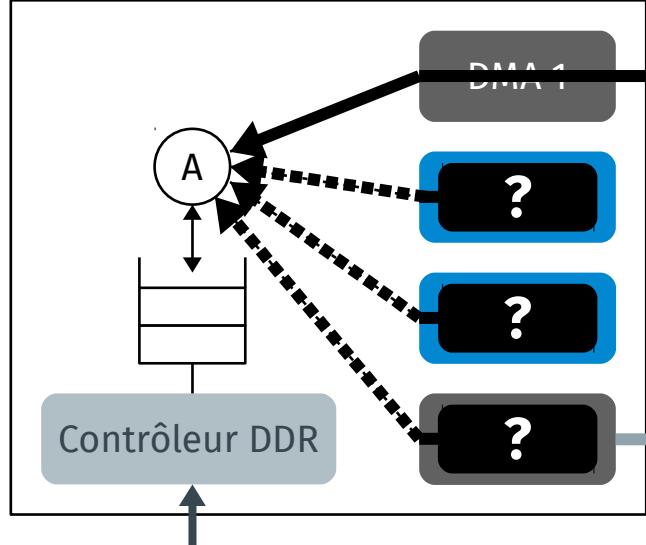


Cluster de calcul 2

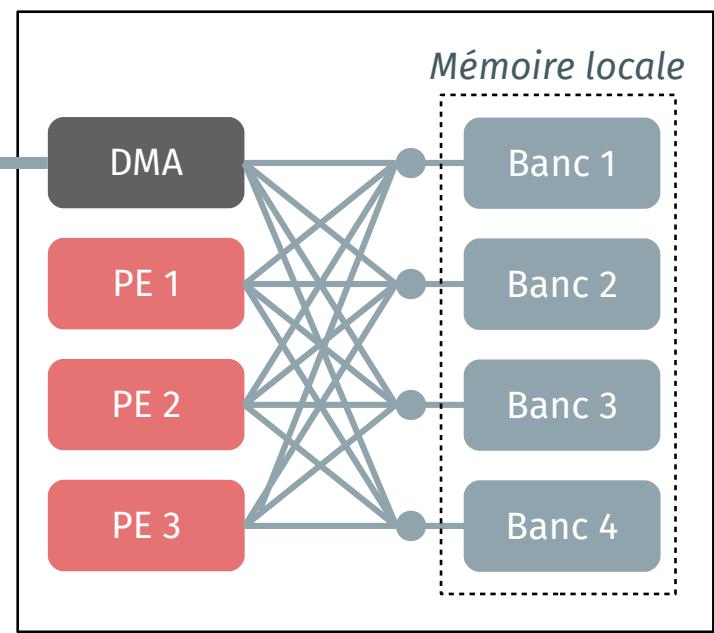


Accès à la DDR-SDRAM

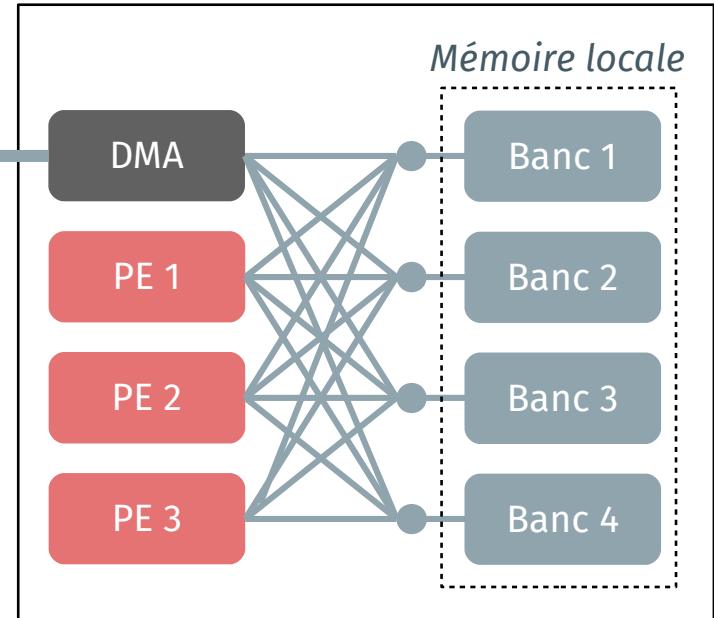
Cluster E/S



Cluster de calcul 1



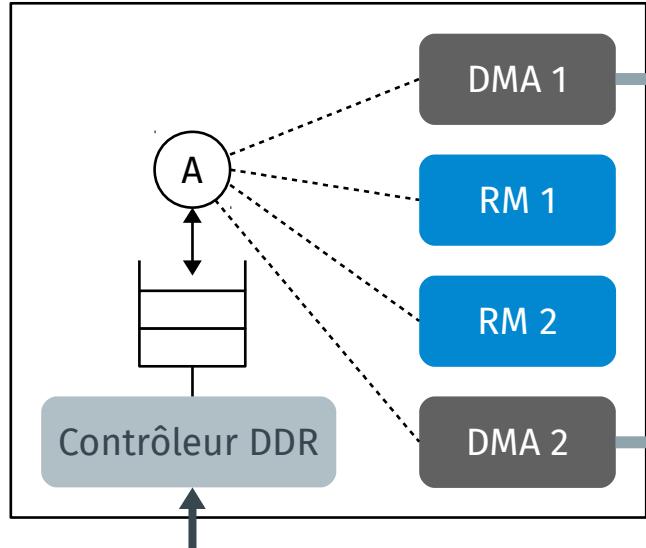
Cluster de calcul 2



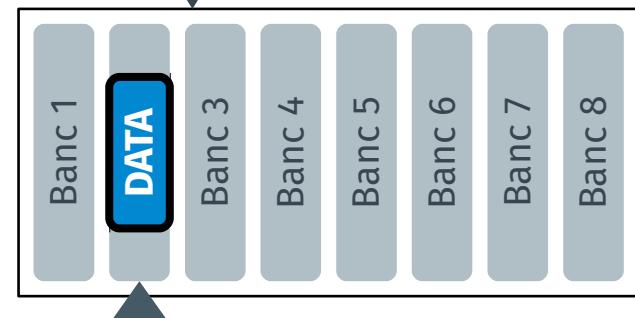
Destination

Accès à la DDR-SDRAM

Cluster E/S

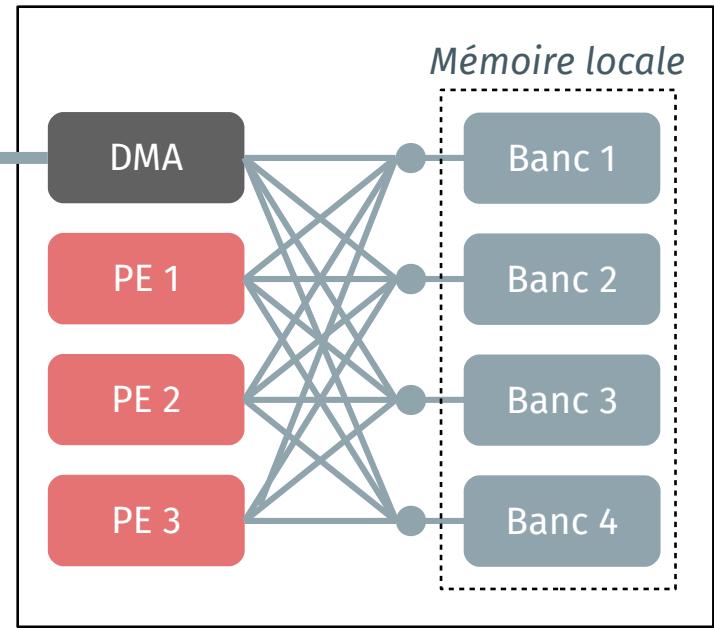


DDR-SDRAM

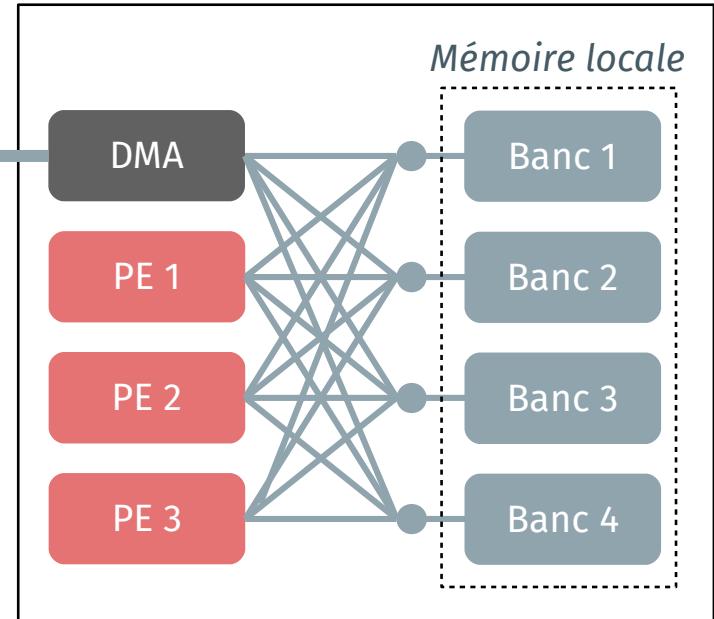


Destination

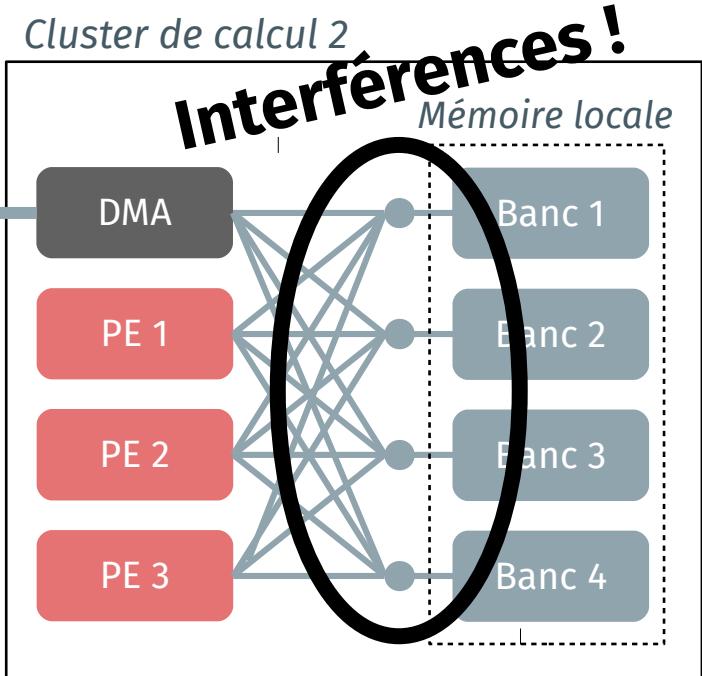
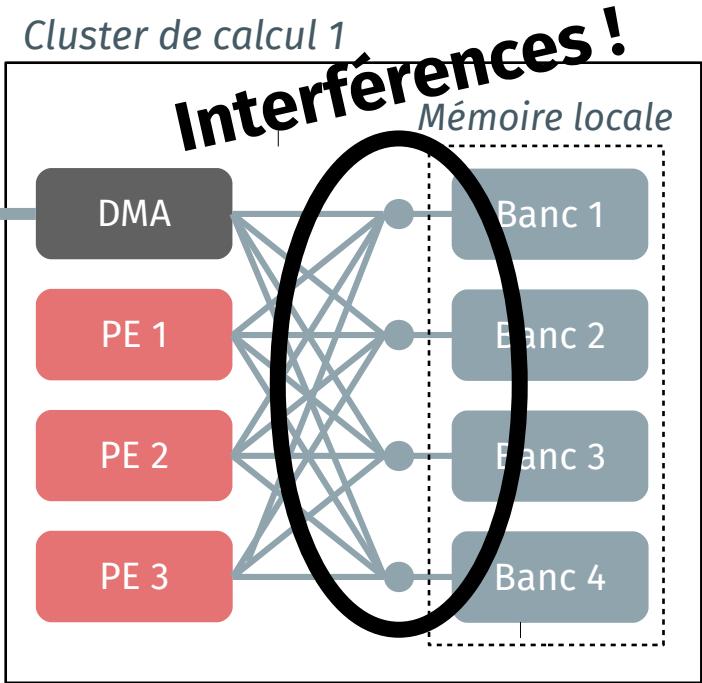
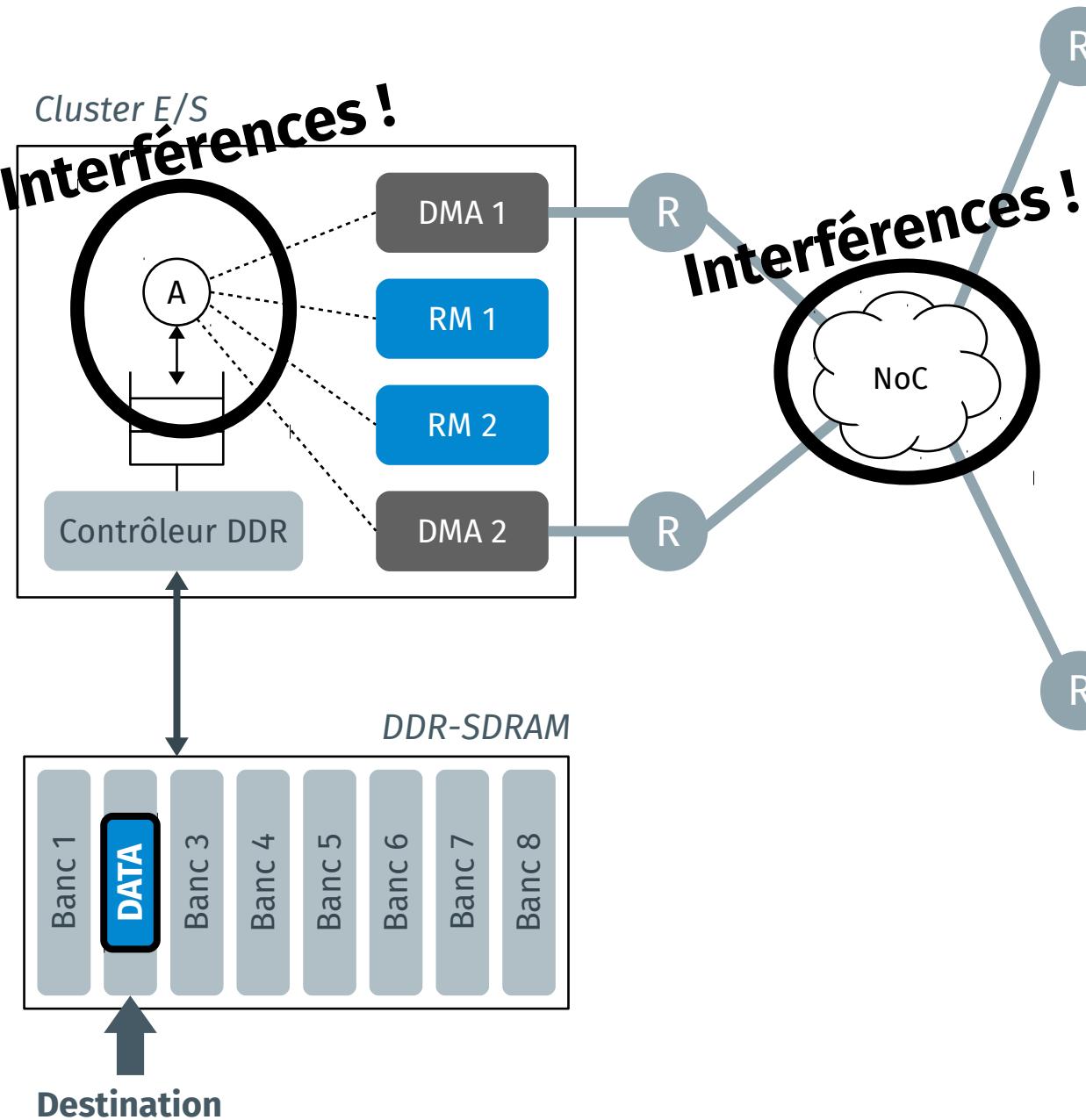
Cluster de calcul 1



Cluster de calcul 2



Accès à la DDR-SDRAM



Travaux antérieurs

- Approche par ***logiciel de contrôle***
 - Deterministic Adaptive Scheduling [1] par Fischer
 - MARTHY [2] par Jean *et al.*
 - Memguard [3] par Yun *et al.*

Travaux antérieurs

- Approche par ***logiciel de contrôle***
 - Deterministic Adaptive Scheduling [1] par Fischer
 - MARTHY [2] par Jean *et al.*
 - Memguard [3] par Yun *et al.*
- Approche par ***modification des applications***
 - Predictable Execution Model [4] par Pellizzoni *et al.*
 - Acquisition-Execution-Restitution [5] par Durrieu *et al.*
 - Scratchpad-Centric Approach [6] par Tabish *et al.*

Modèle d'exécution et partitions

Définition

“Un **modèle d'exécution** est un ensemble de règles contraignant le comportement des applications afin d'éviter des effets indésirables.”

Modèle d'exécution et partitions

Définition

“Un **modèle d'exécution** est un ensemble de règles contraignant le comportement des applications afin d'éviter des effets indésirables.”

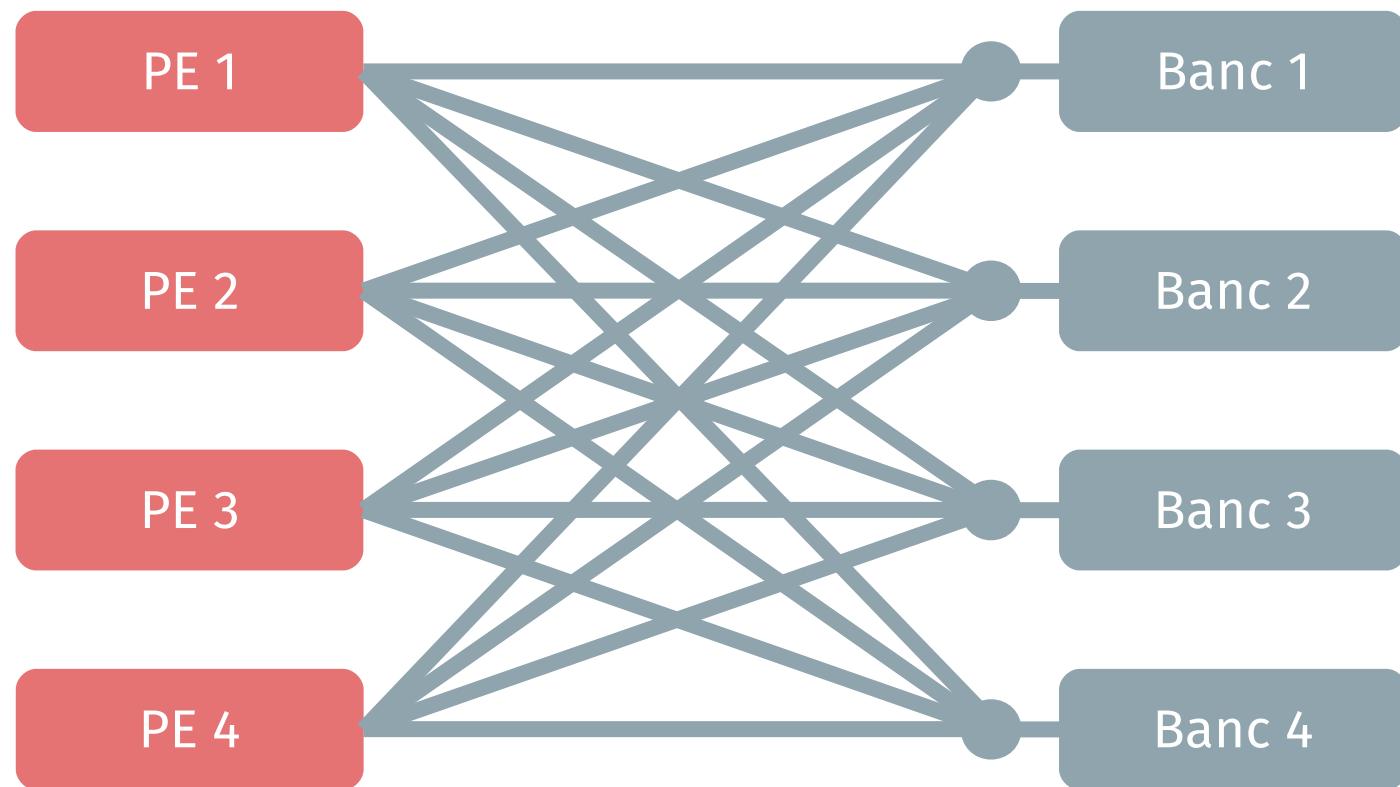
Définition

“Une **partition** est un environnement d'exécution isolé garantissant à une application qu'elle ne subira pas d'interférences extérieures à sa partition.”

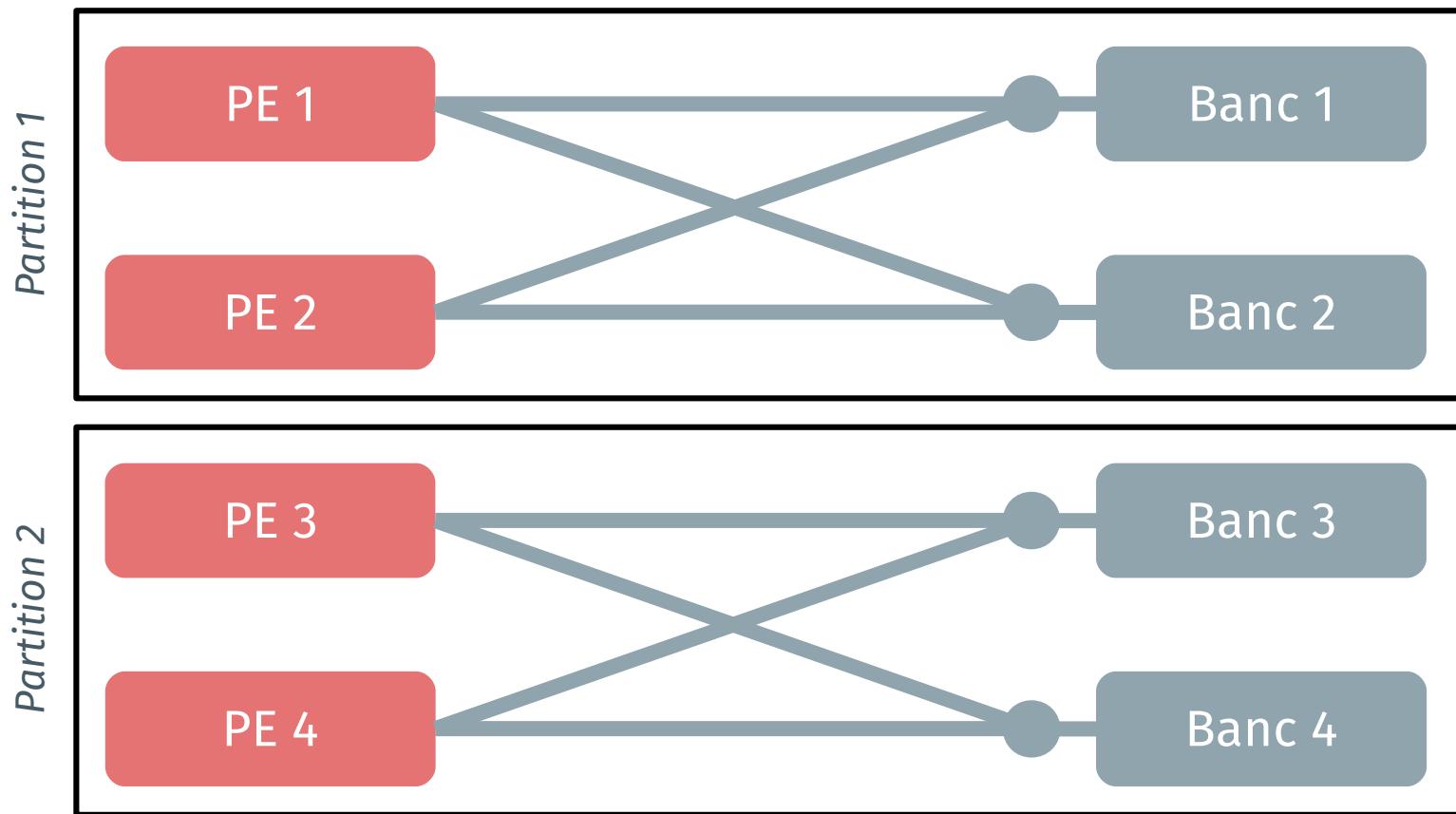
Règle 1.

Dans un cluster de calcul, ***chaque banc de mémoire locale et chaque PE*** n'est utilisable que par ***une seule partition***

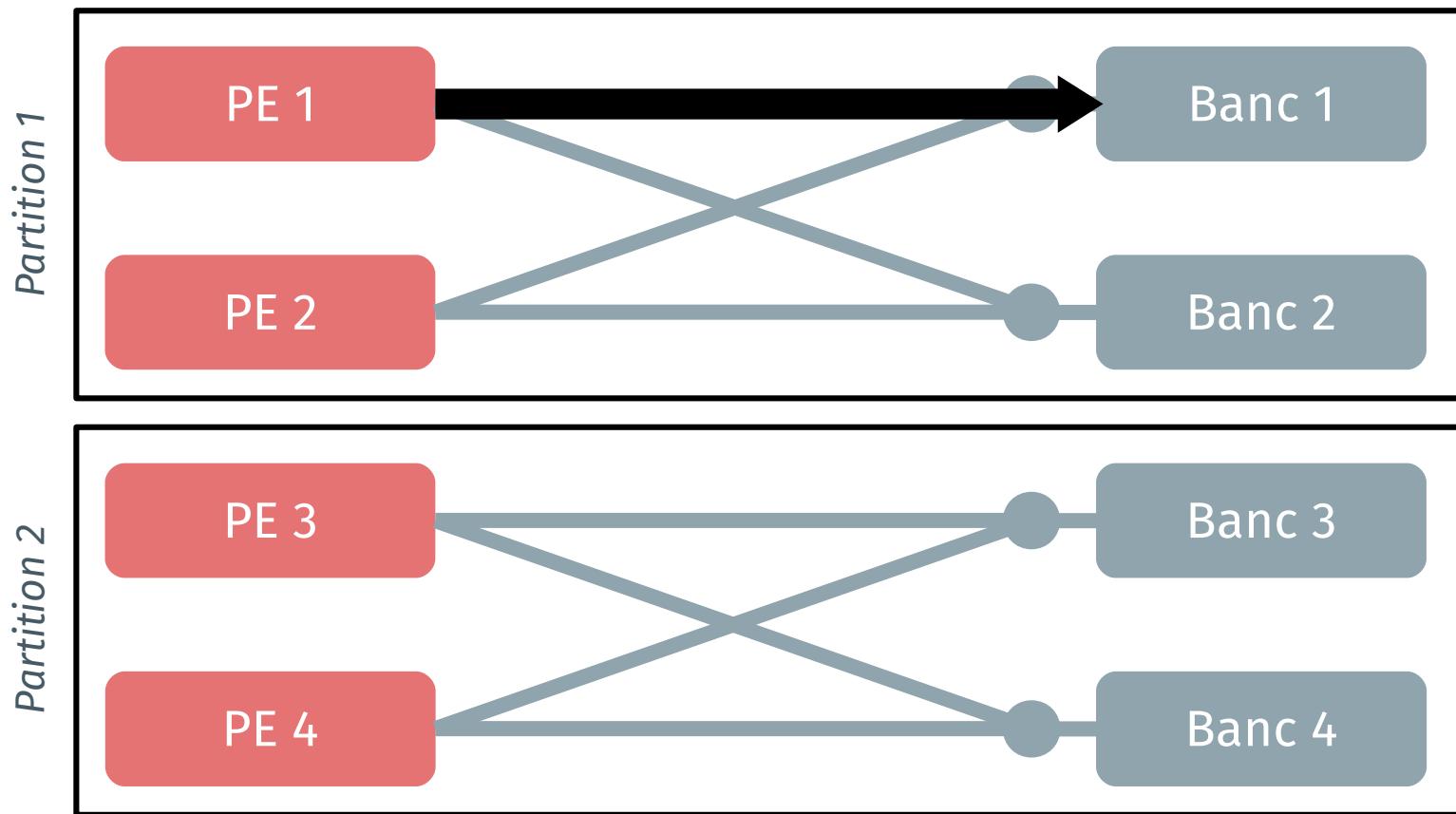
Règle 1



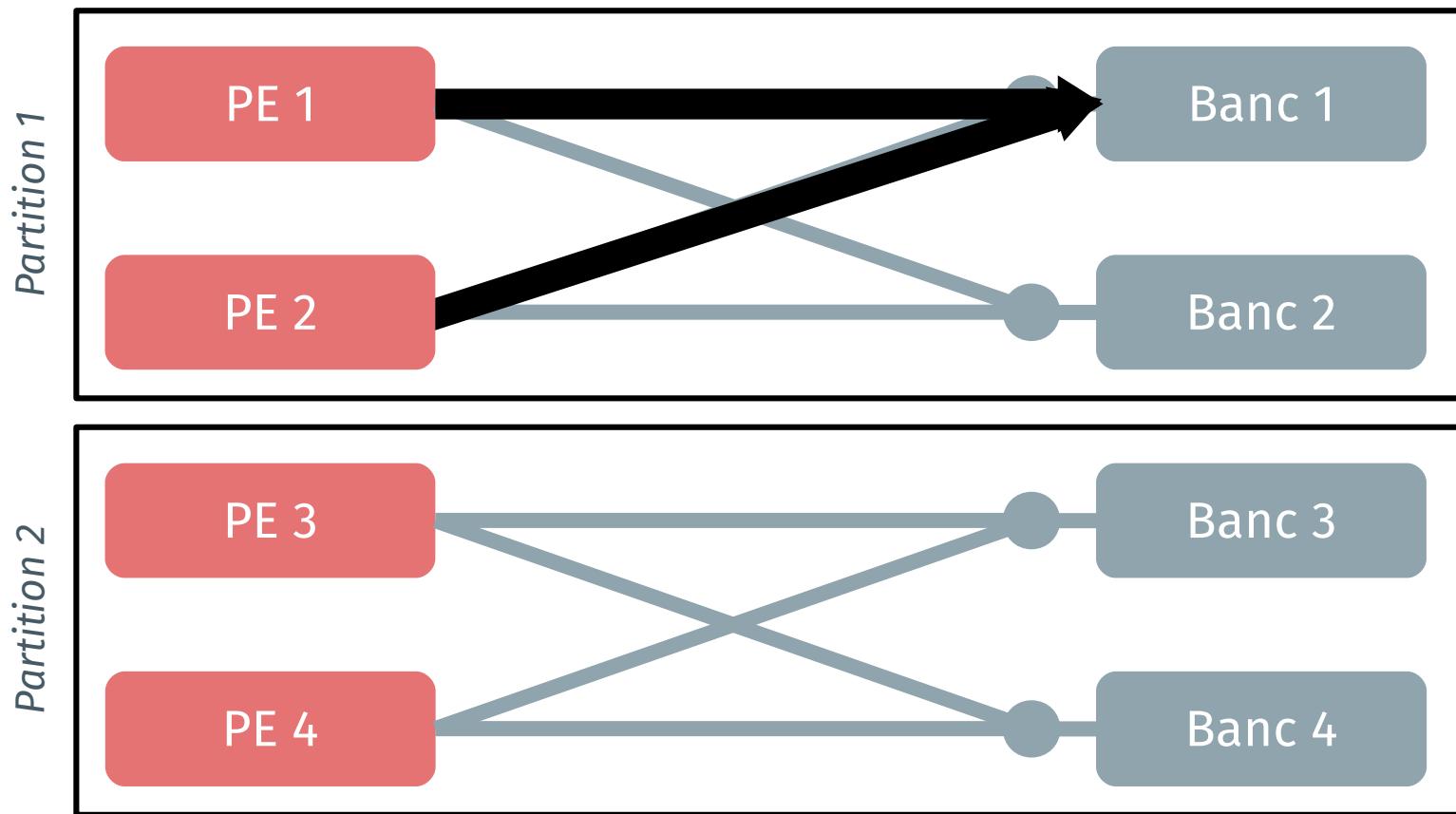
Règle 1



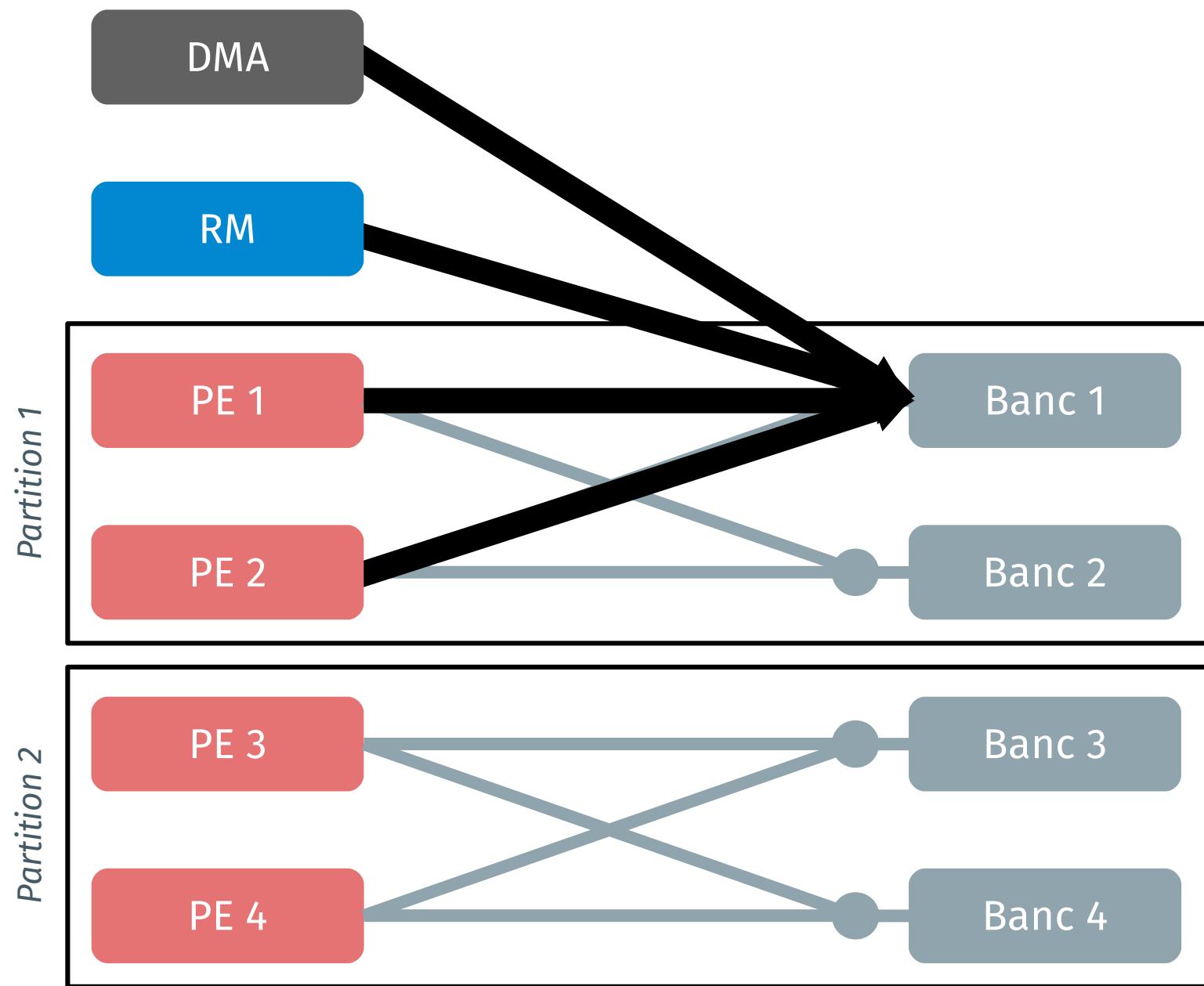
Règle 1



Règle 1



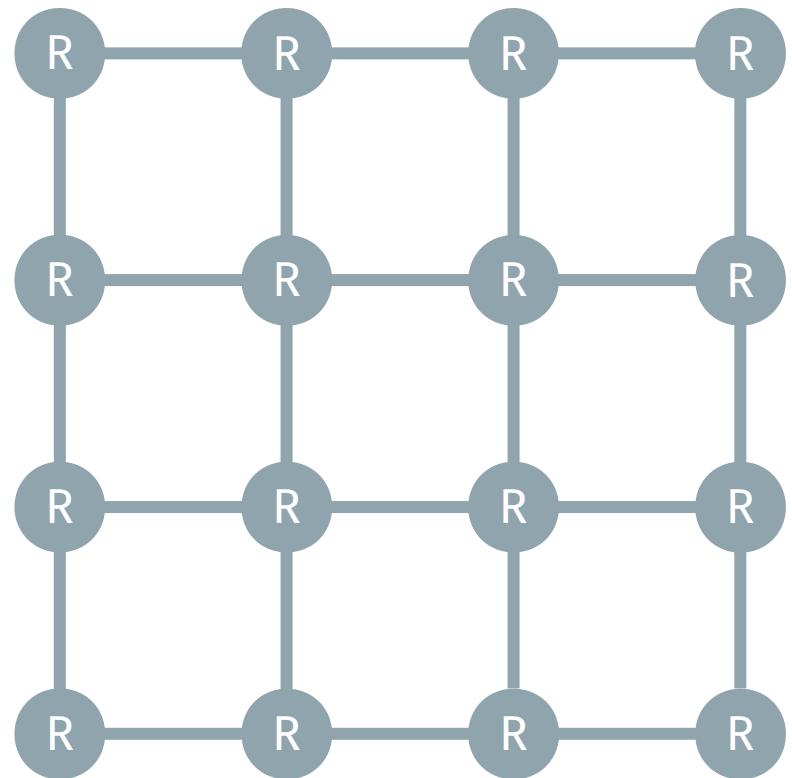
Règle 1



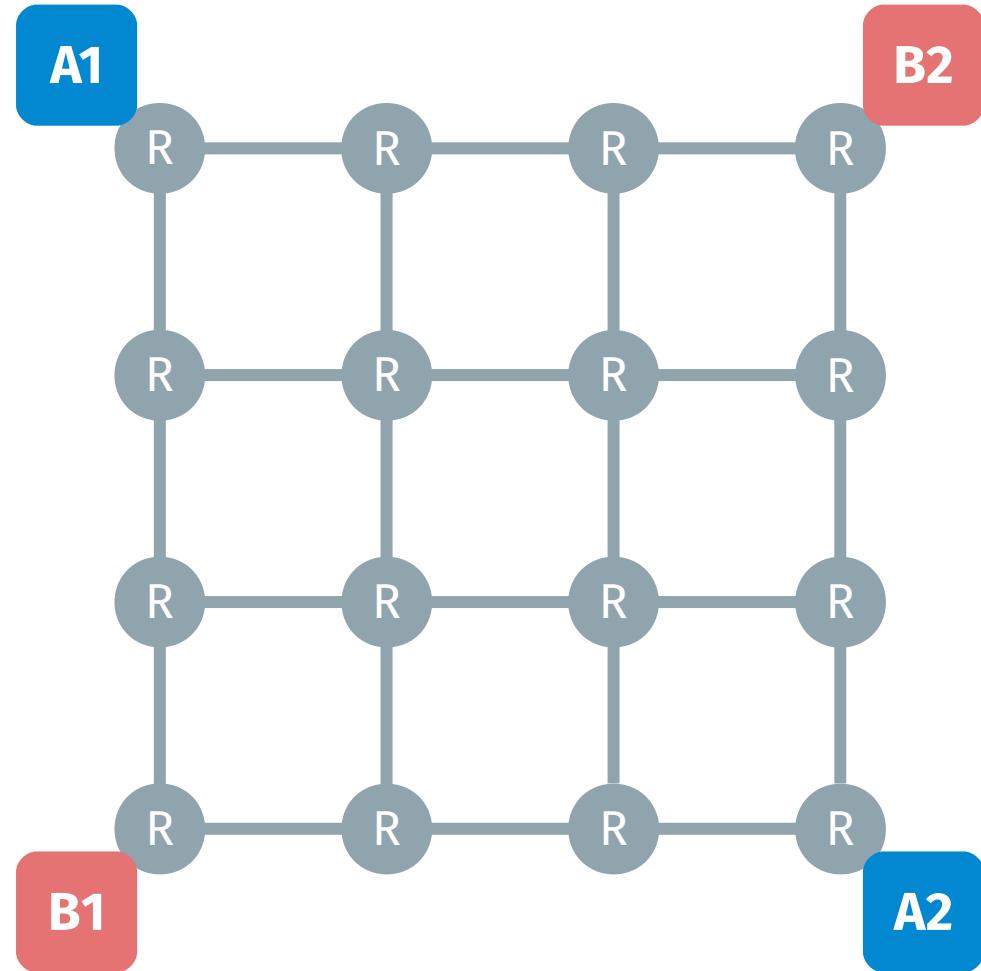
Règle 2.

L'ordonnancement du NoC est **global** et **dirigé par le temps**. Les conflits d'accès aux liens sont résolus **par construction**.

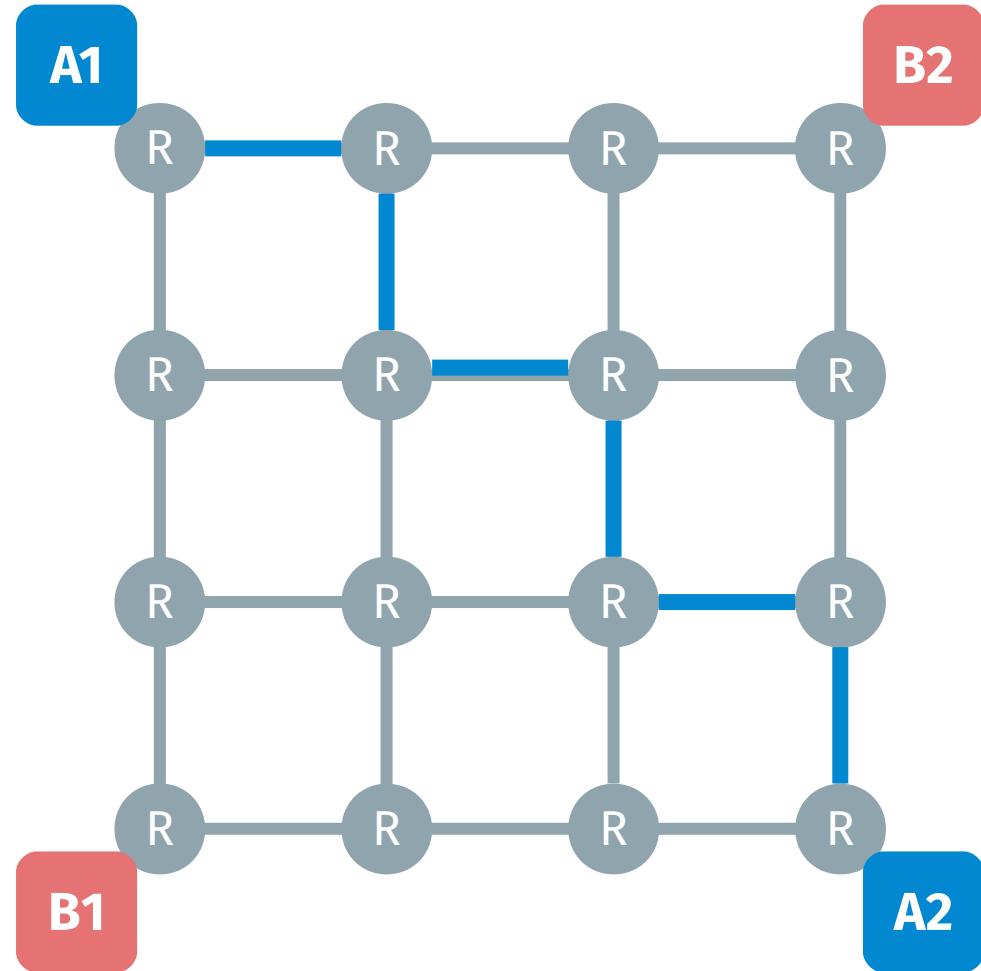
Règle 2



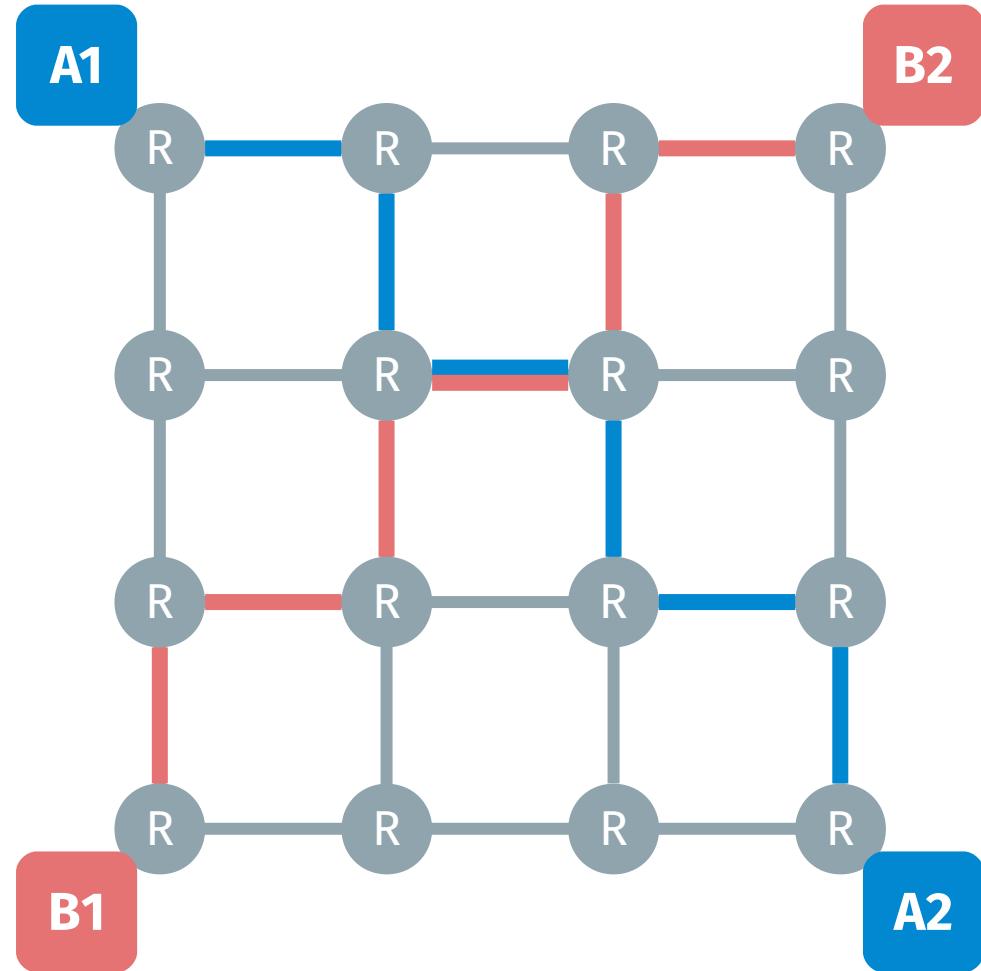
Règle 2



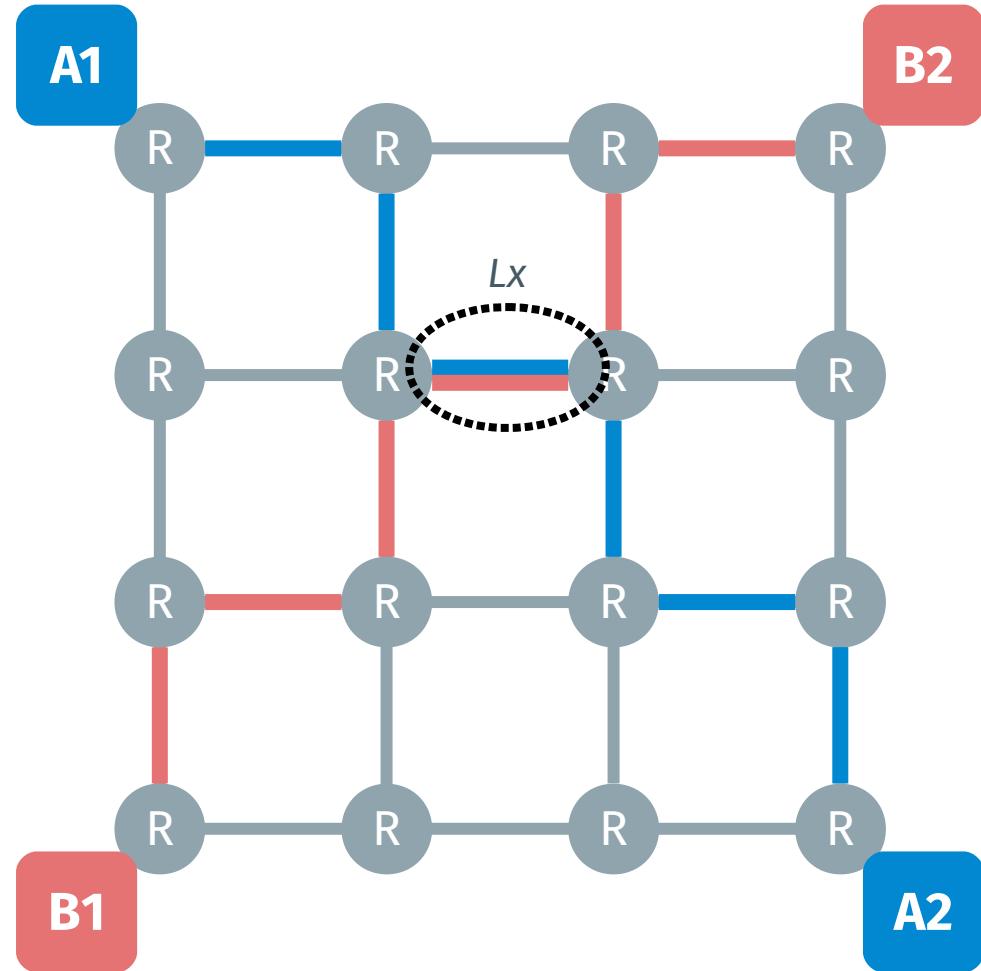
Règle 2



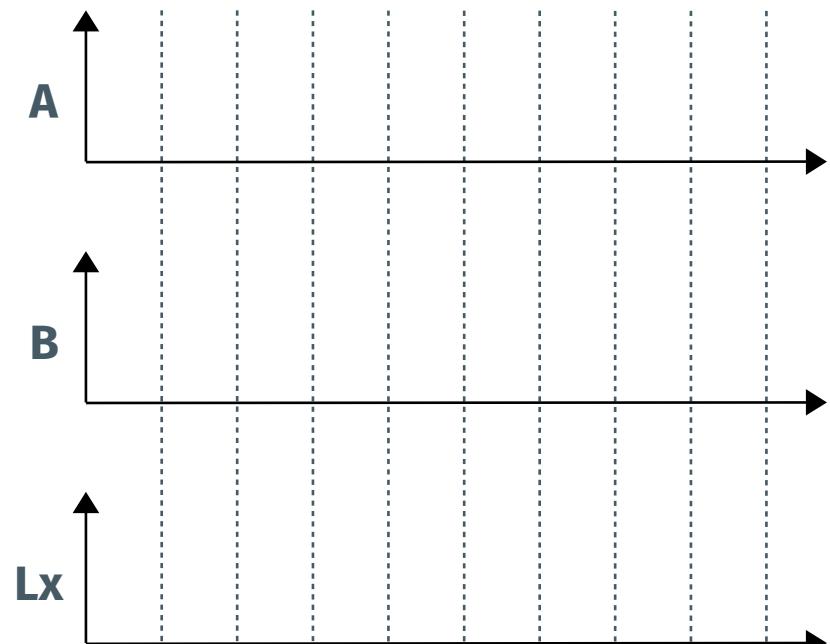
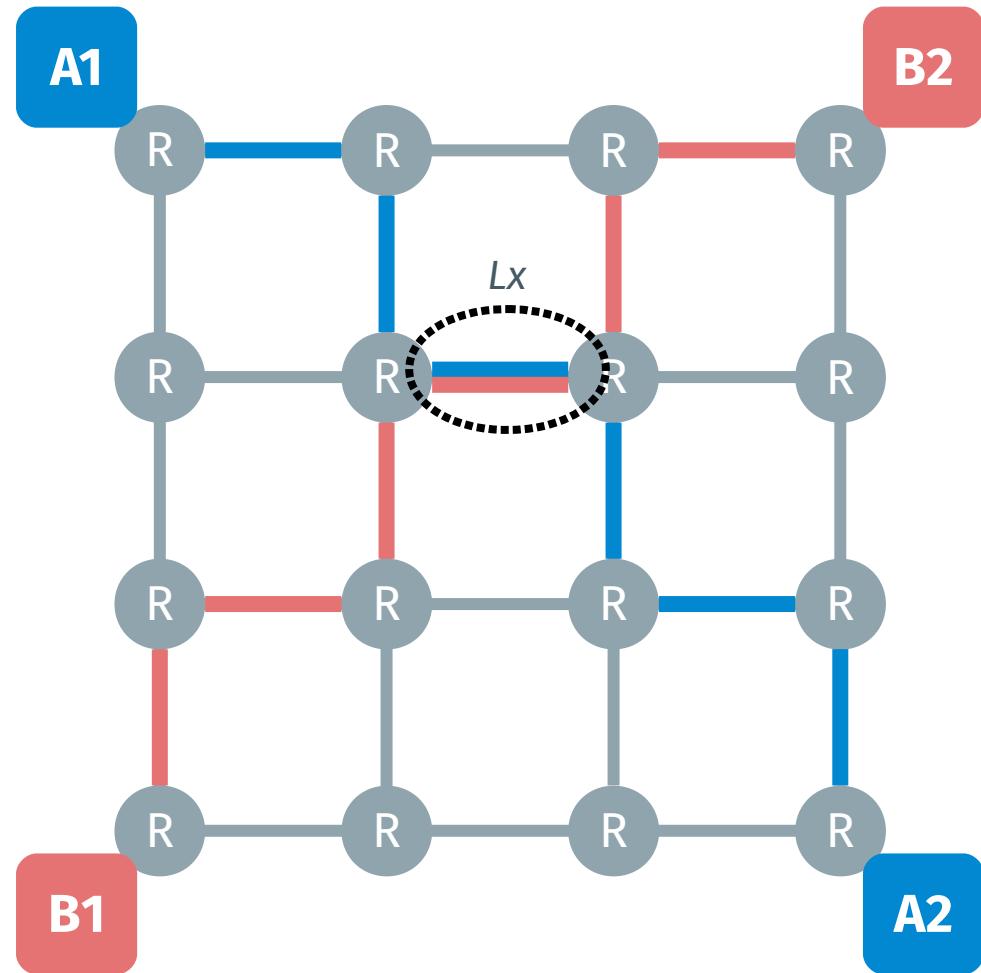
Règle 2



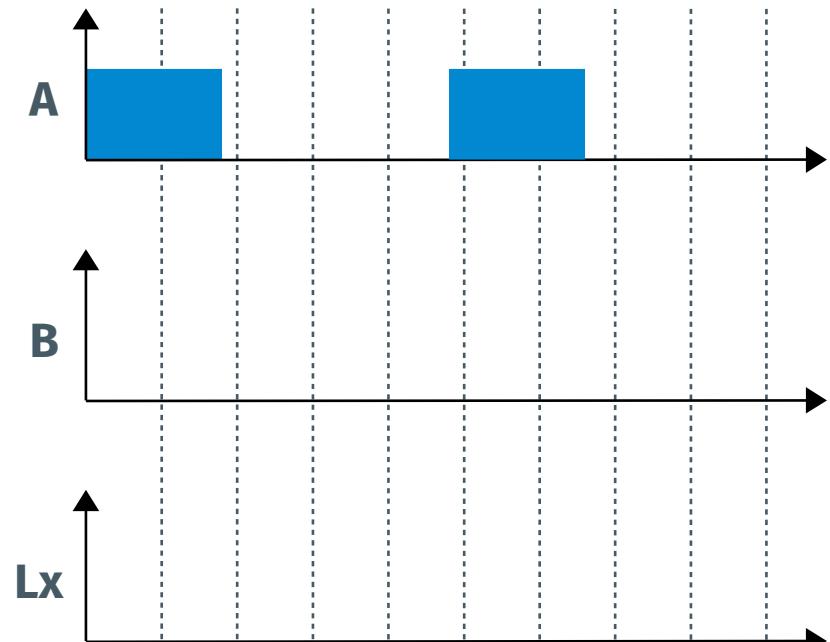
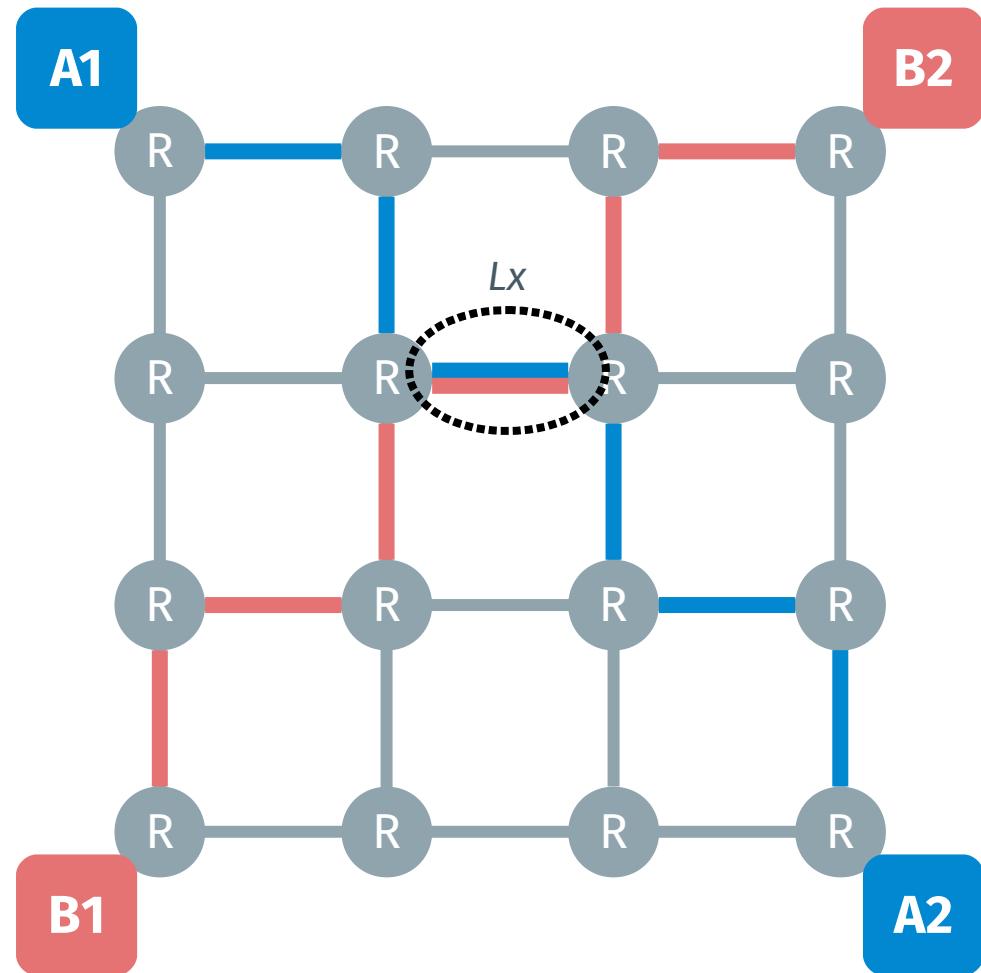
Règle 2



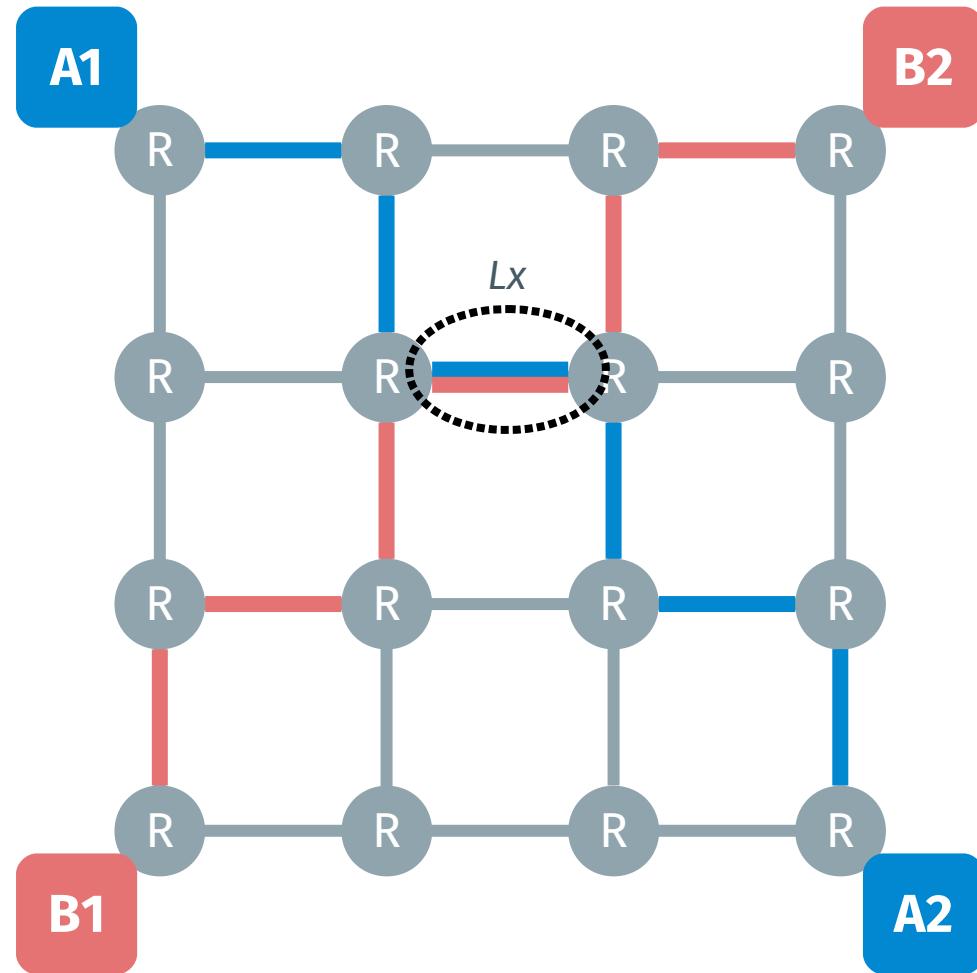
Règle 2



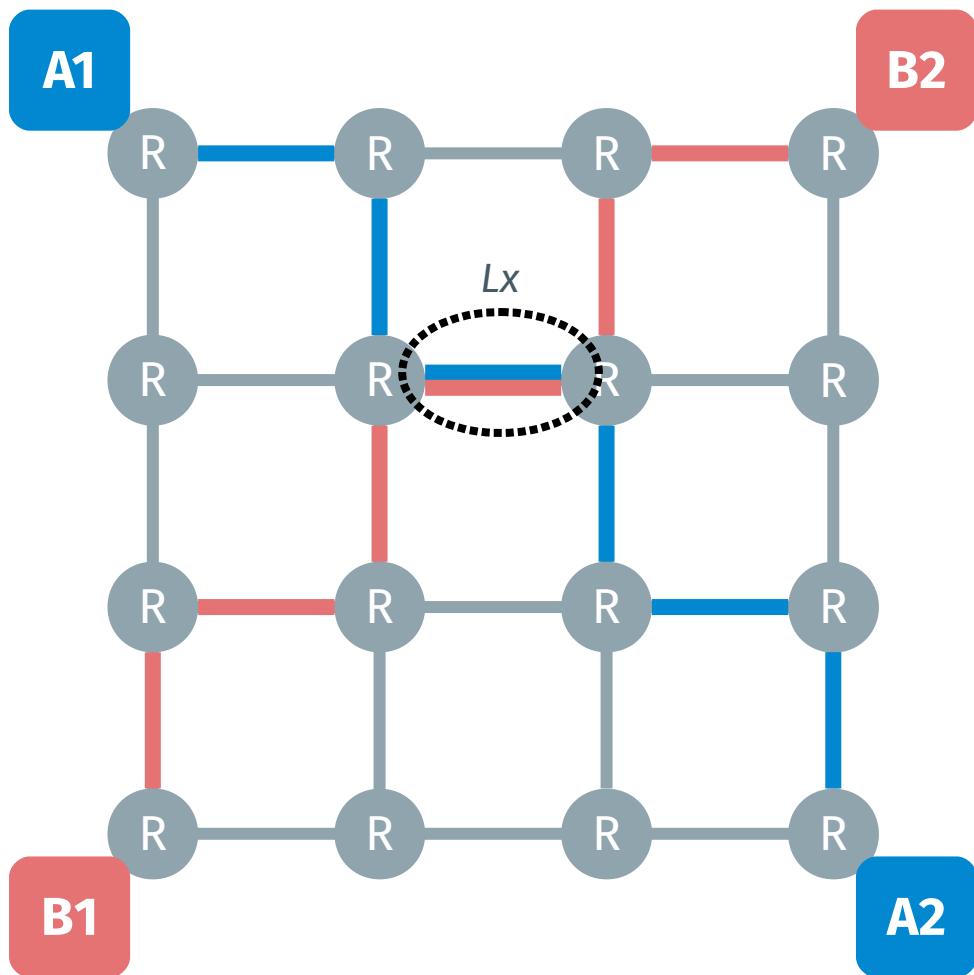
Règle 2



Règle 2



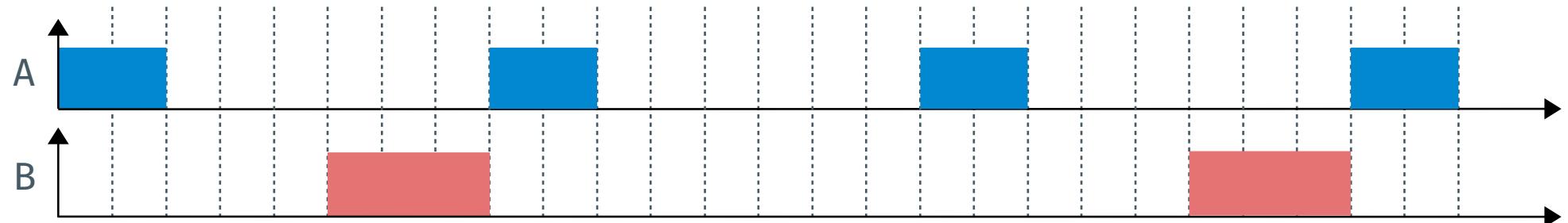
Règle 2



Règle 3.

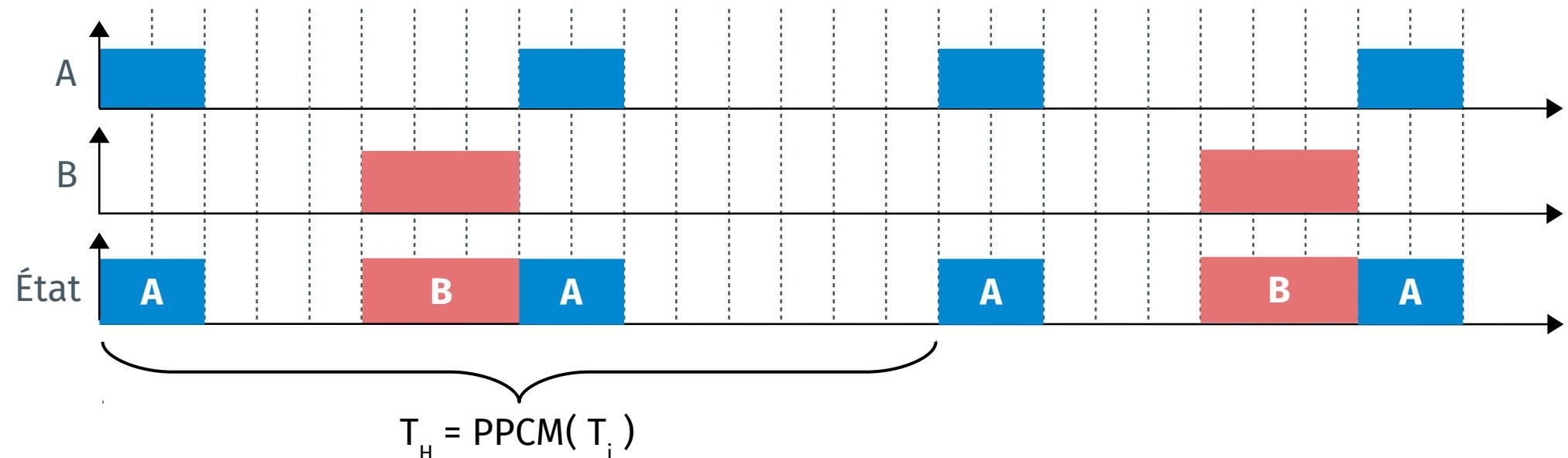
Les ***zones mémoires*** à envoyer par le NoC
doivent être connues ***hors-ligne***.

Règle 3



	T _i	O _i
A	8	0
B	16	5

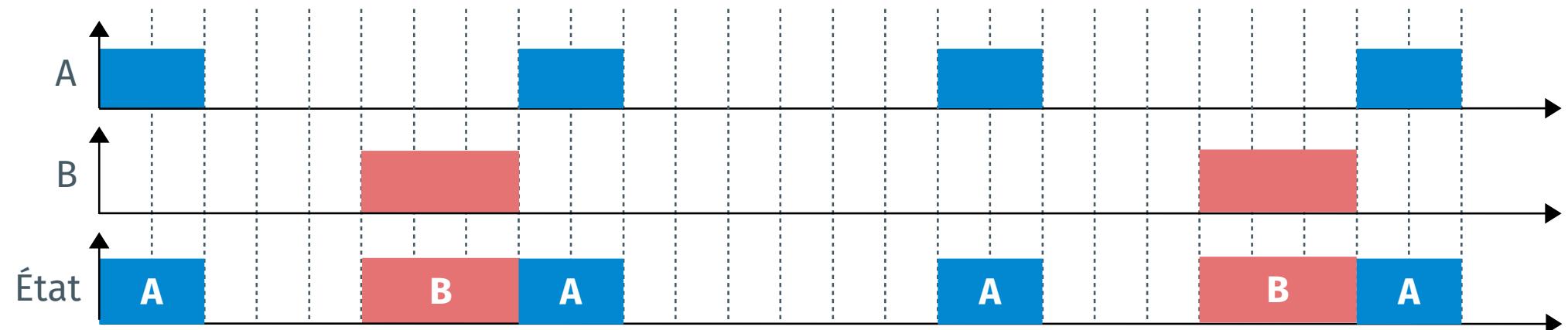
Règle 3



	T_i	O_i
A	8	0
B	16	5

État	A	Vide	B	A	Vide
Durée	2	3	3	2	6

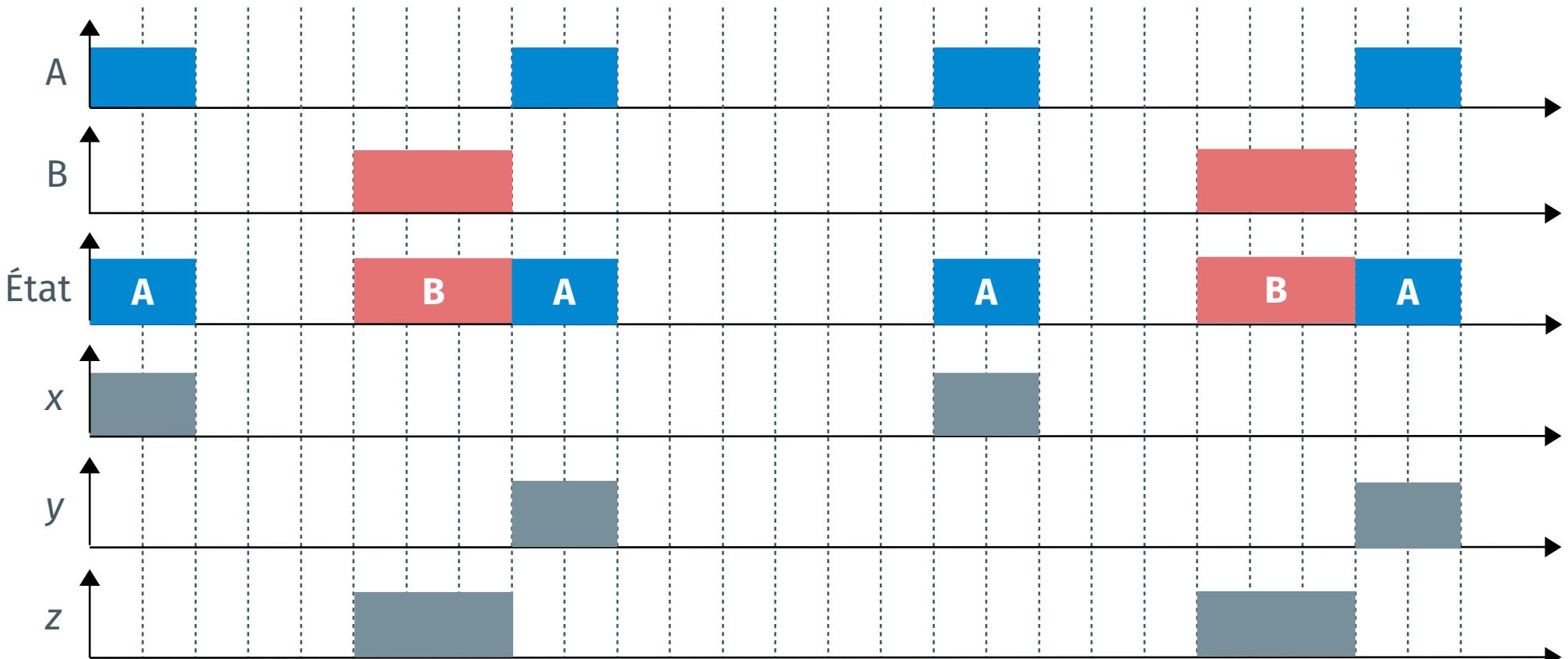
Règle 3



	T_i	O_i
A	8	0
B	16	5

État	A	Vide	B	A	Vide
Durée	2	3	3	2	6

Règle 3

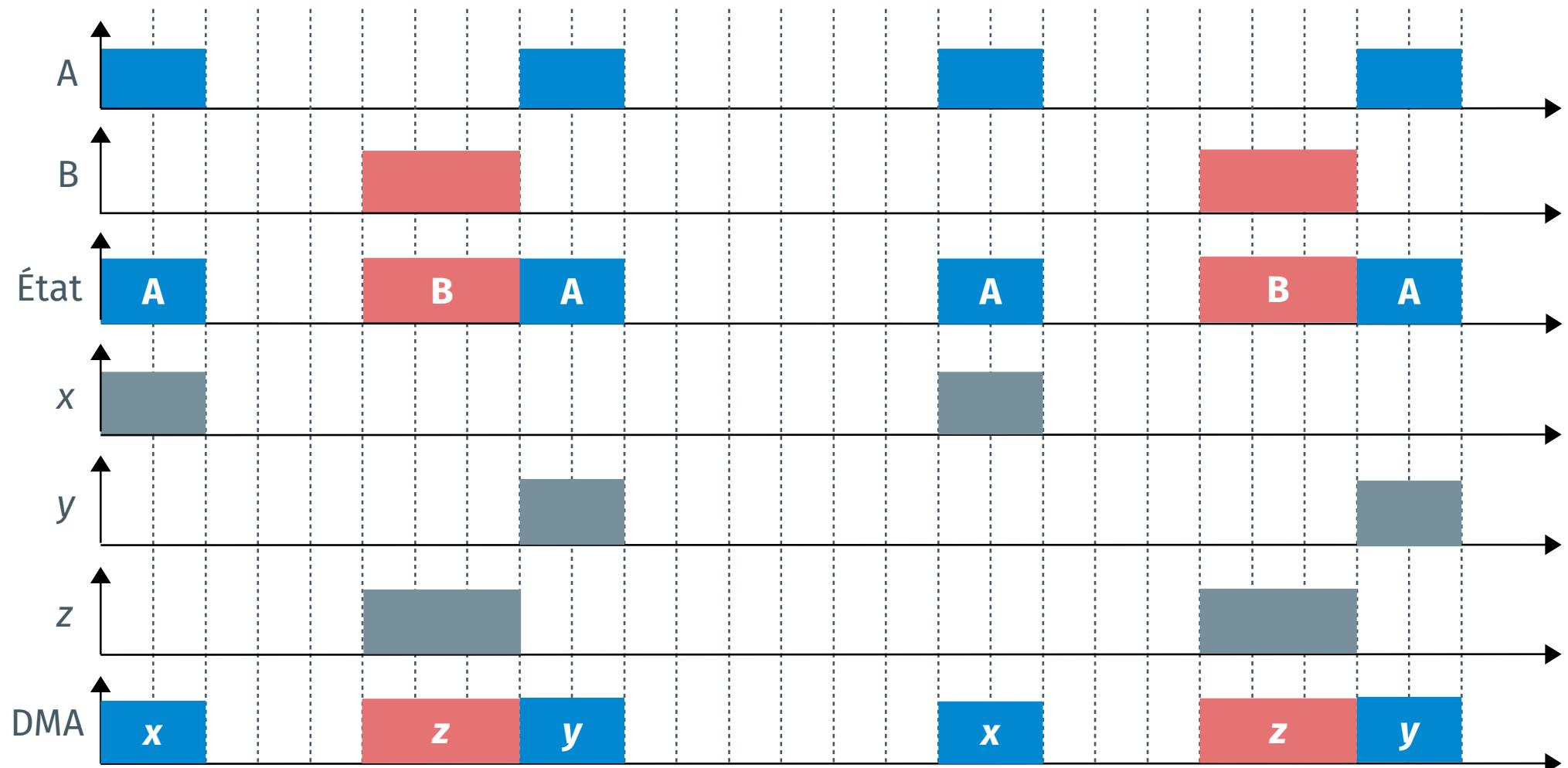


	T_i	O_i
A	8	0
B	16	5

État	A	Vide	B	A	Vide
Durée	2	3	3	2	6

$$\begin{aligned}\mathbf{A} &\rightarrow \{x, y\} \\ \mathbf{B} &\rightarrow \{z\}\end{aligned}$$

Règle 3



	T_i	O_i
A	8	0
B	16	5

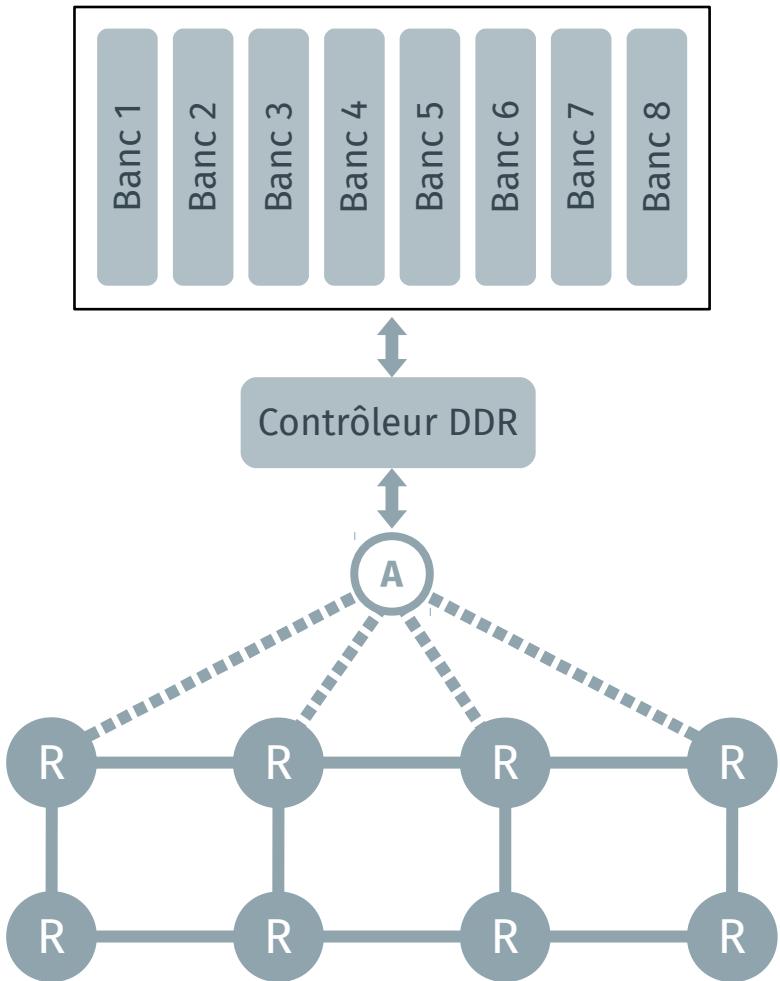
État	A	Vide	B	A	Vide
Durée	2	3	3	2	6

$$\begin{aligned}\mathbf{A} &\rightarrow \{ x, y \} \\ \mathbf{B} &\rightarrow \{ z \}\end{aligned}$$

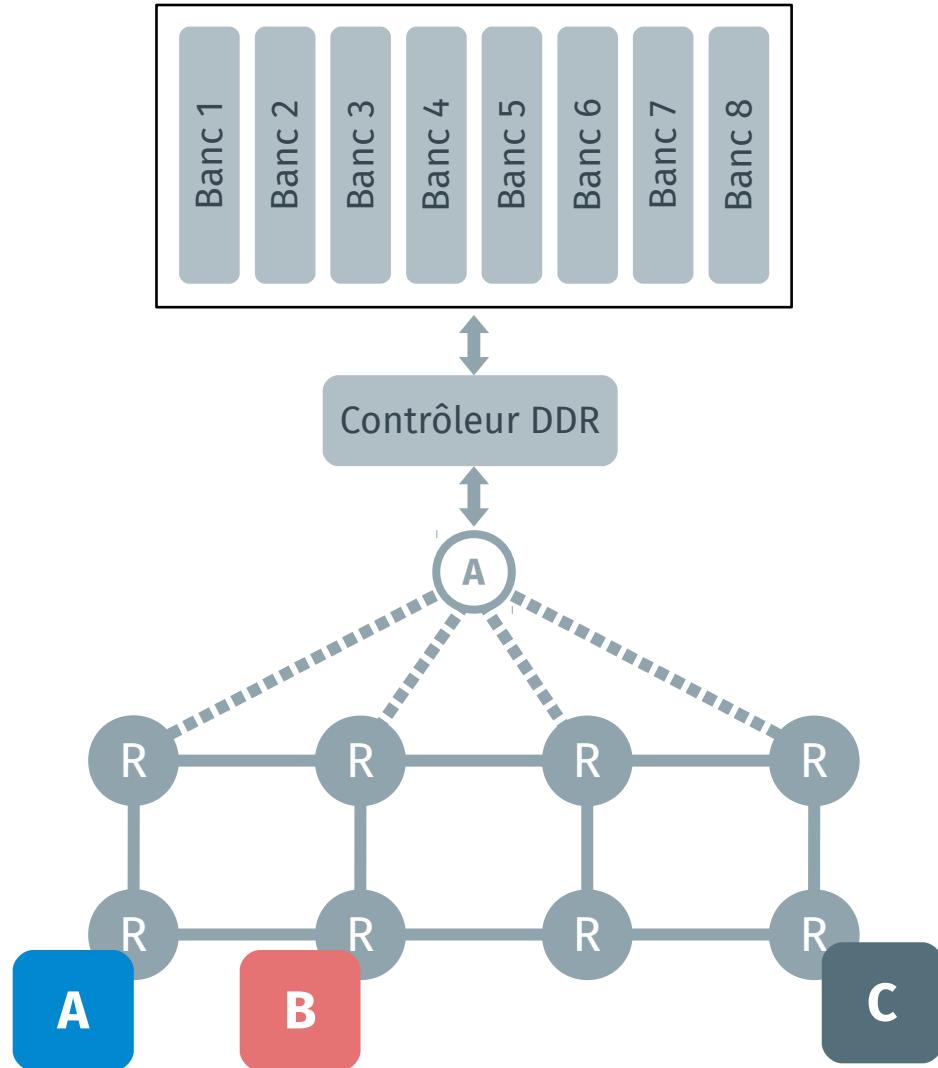
Règle 4.

Un ***banc de DDR-SDRAM*** peut être utilisé par
plusieurs partitions à condition qu'elles
n'y accèdent ***pas simultanément***.

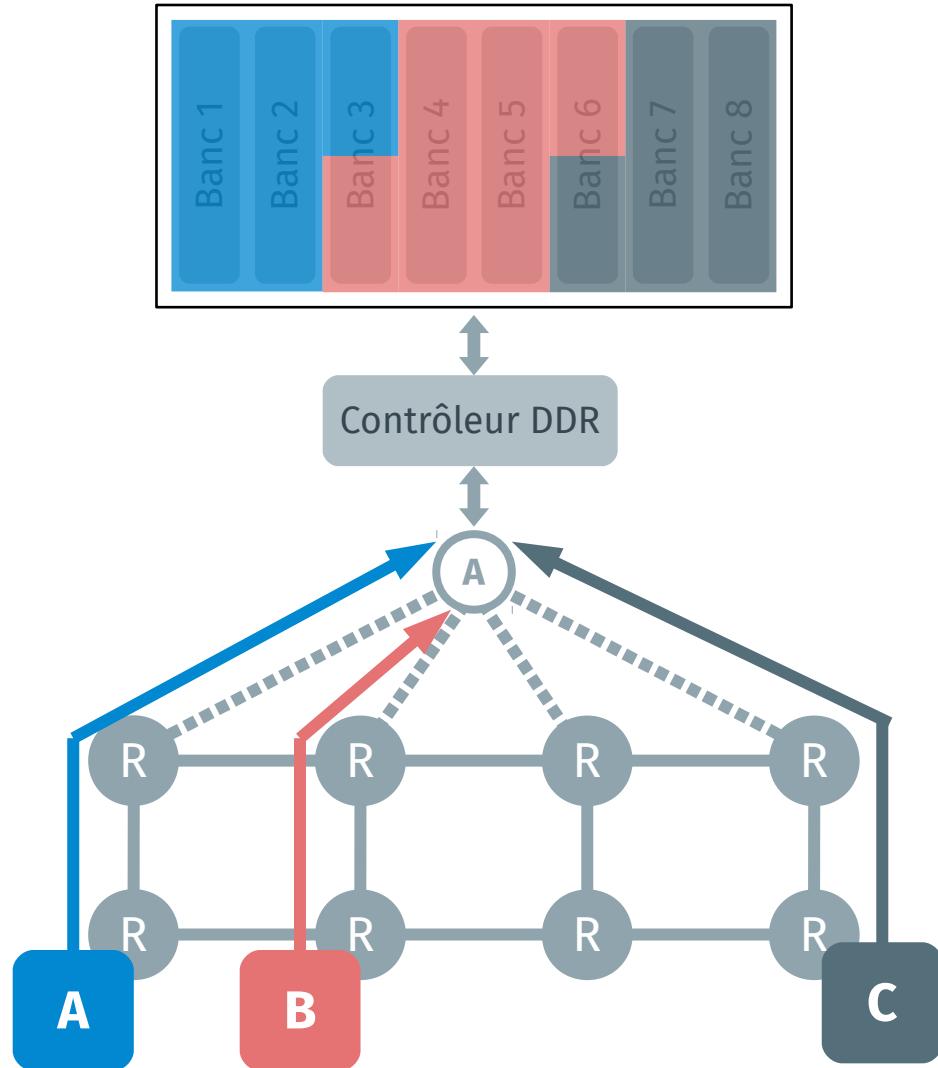
Règle 4



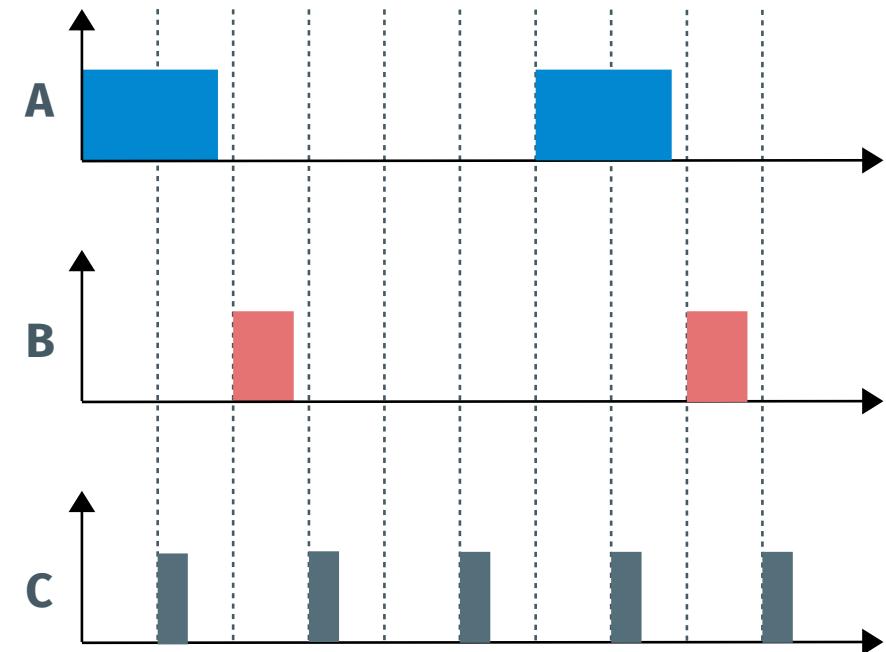
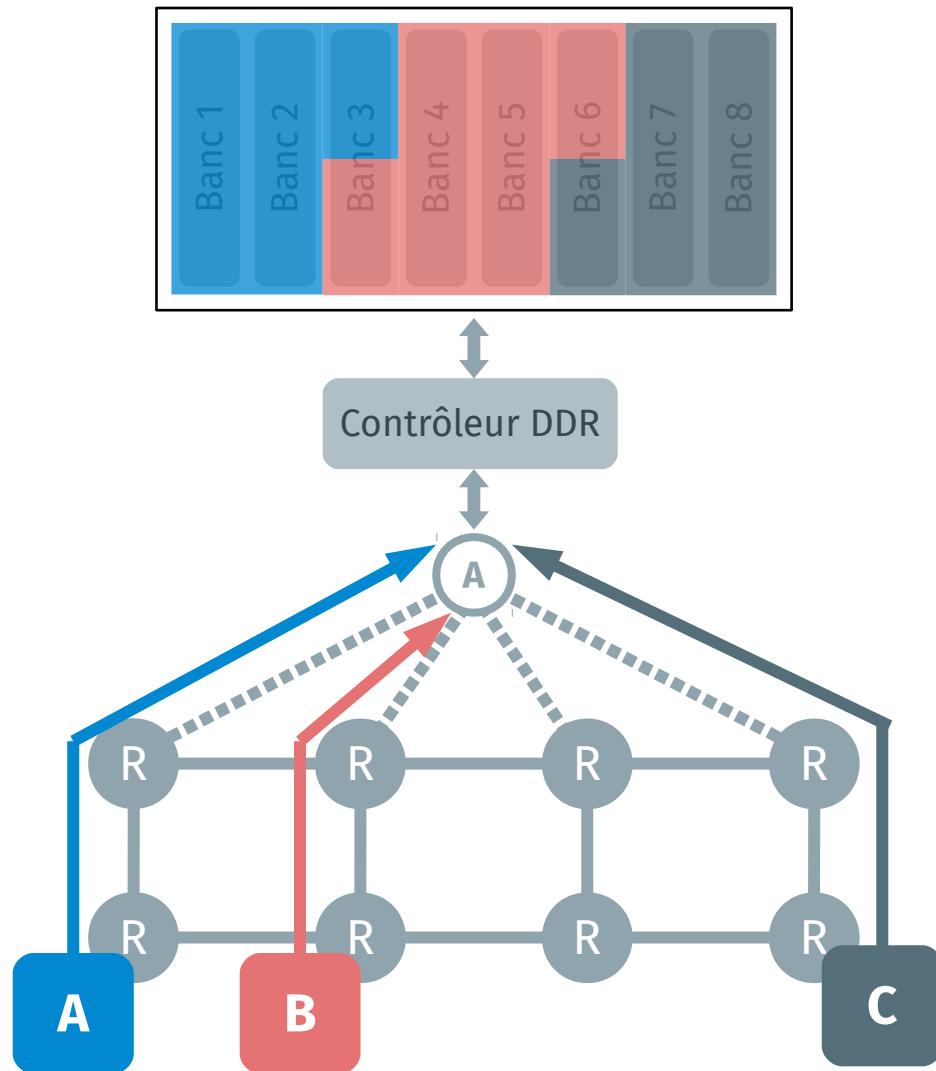
Règle 4



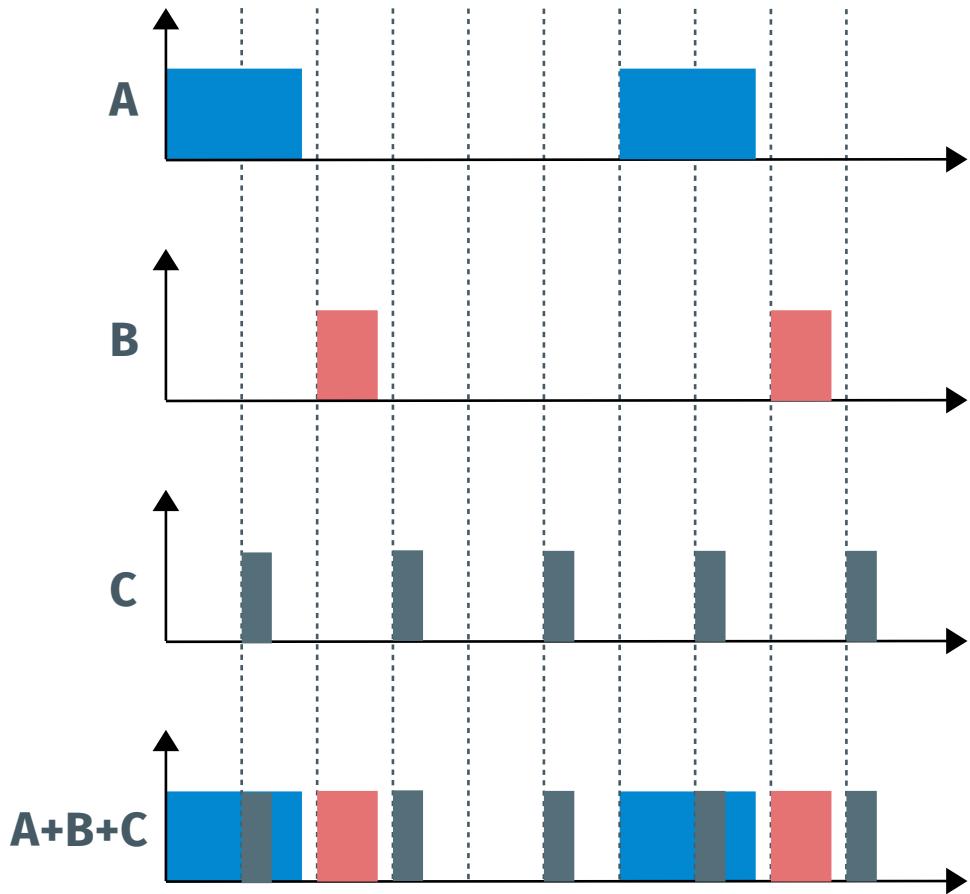
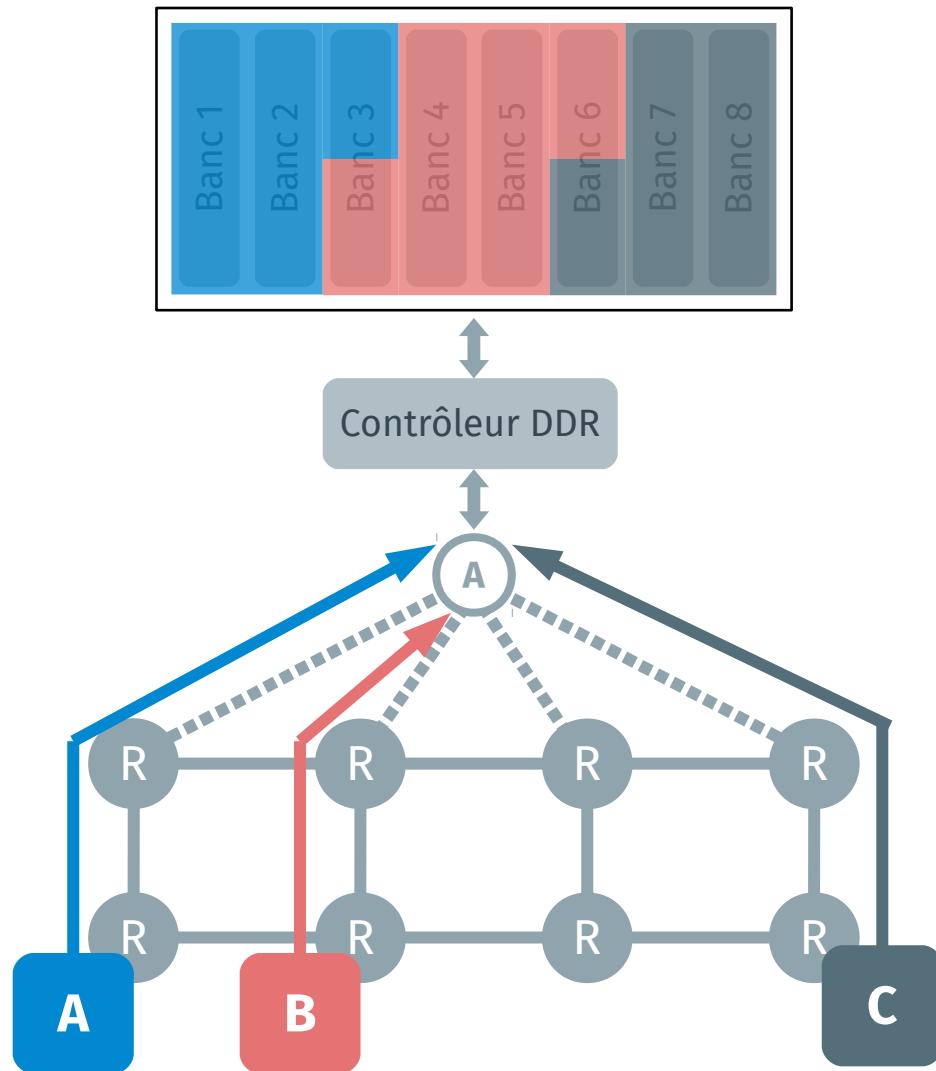
Règle 4



Règle 4



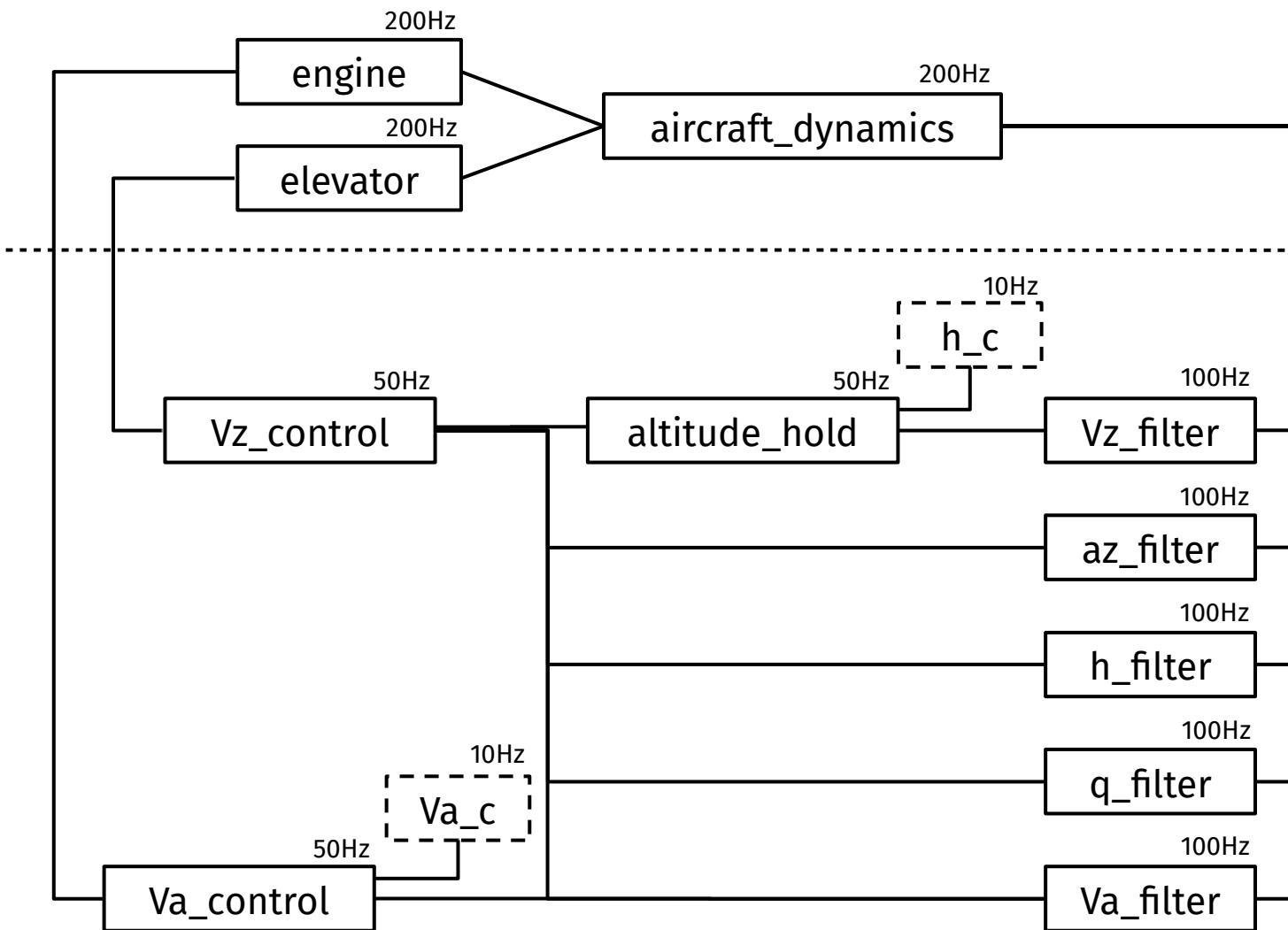
Règle 4



Hyperviseur

- Assure à lui seul le respect des règles du modèle d'exécution ***en-ligne***
- Programmé en ***bare-metal*** pour une meilleure maîtrise
- Exécuté sur le RM de chaque cluster
 - Fixe les ***configurations matérielles***
 - Gère les transactions ***DMA*** et ***l'ordonnancement NoC***
- Code privilégié des PEs
 - ***Code de boot*** spécifique
 - Configuration ***MMUs gelées***
 - ***Handlers*** d'exceptions (Syscalls, traps, ...)

Cas d'étude ROSACE [7]



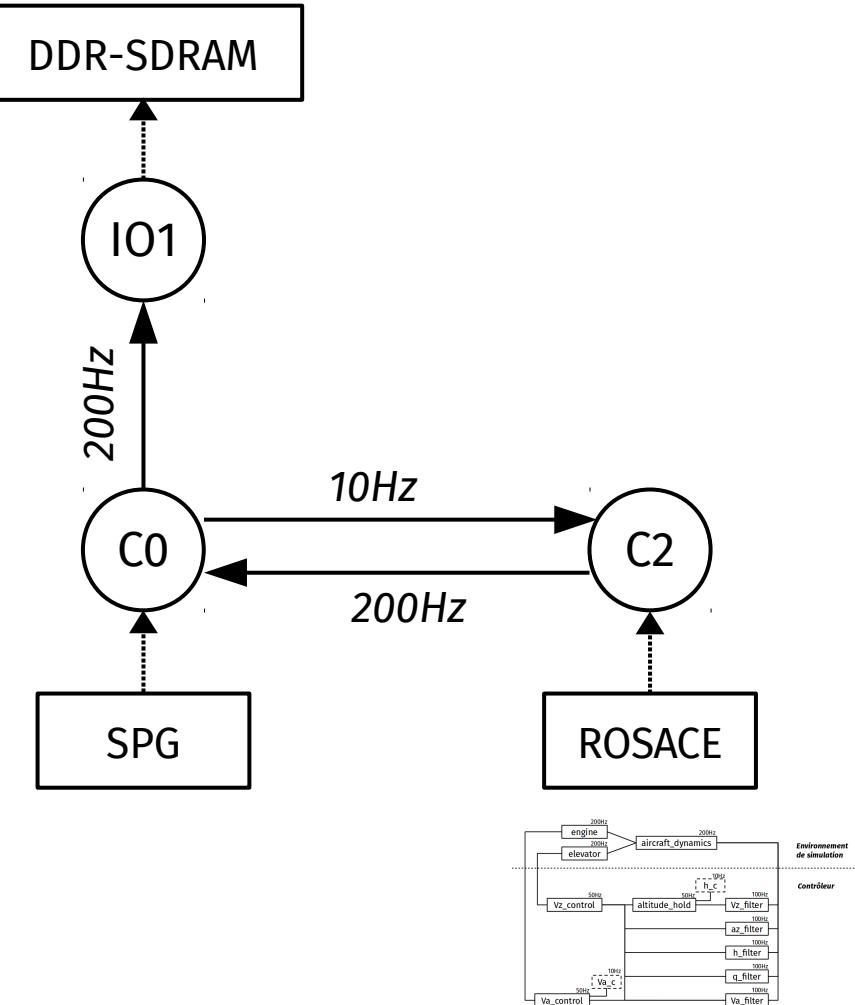
**Environnement
de simulation**

Contrôleur

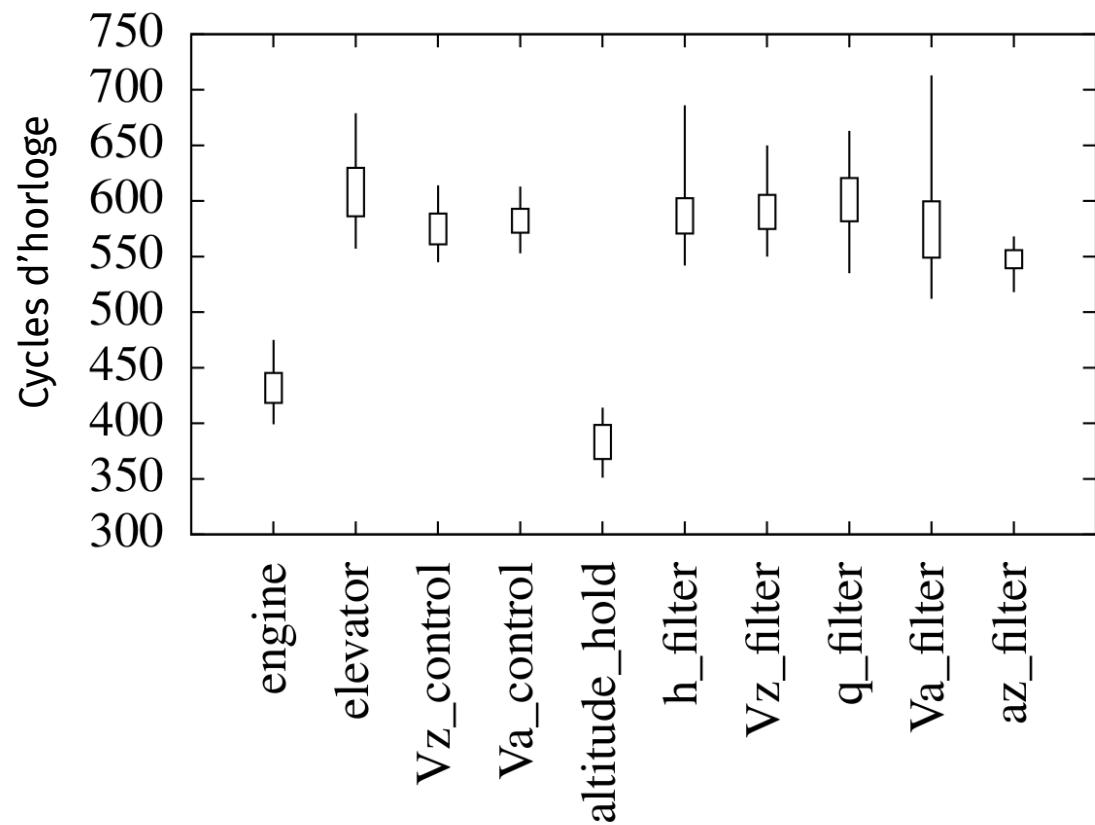
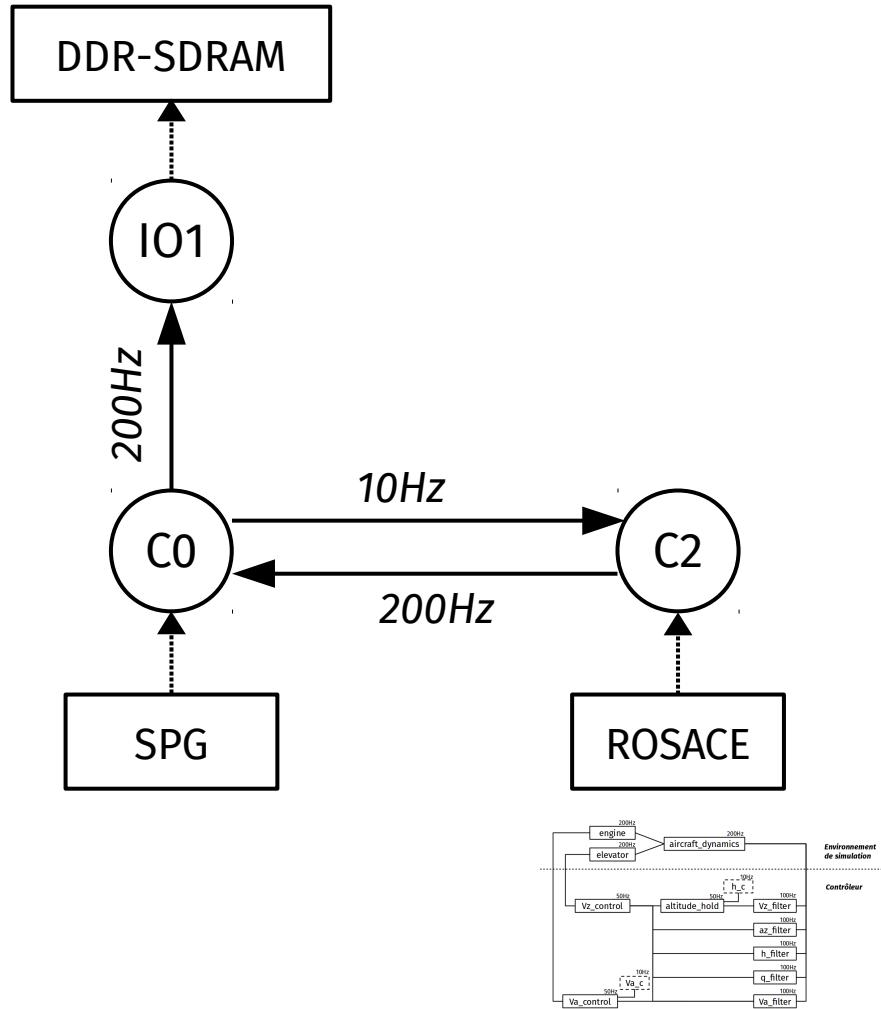
Application concurrente : inversion d'image



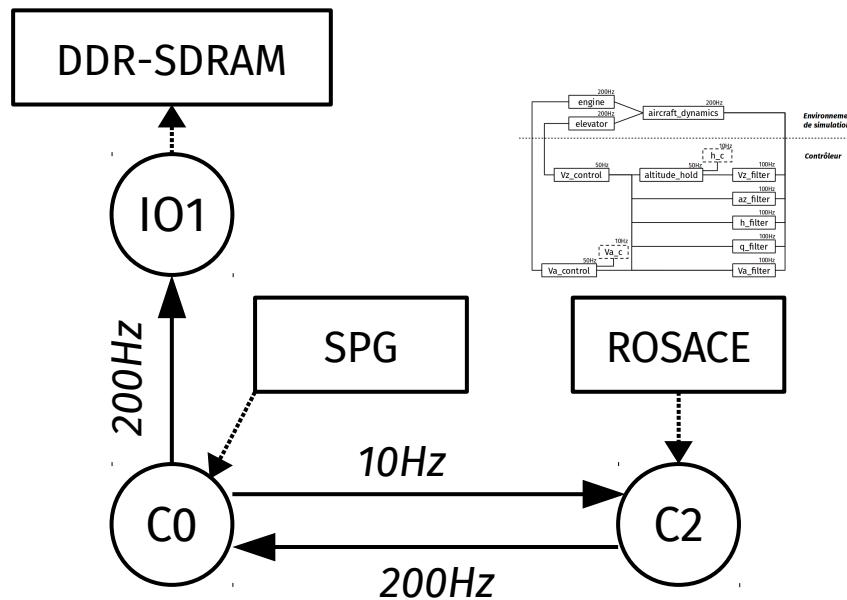
Scénario 1 : pas d'interférences



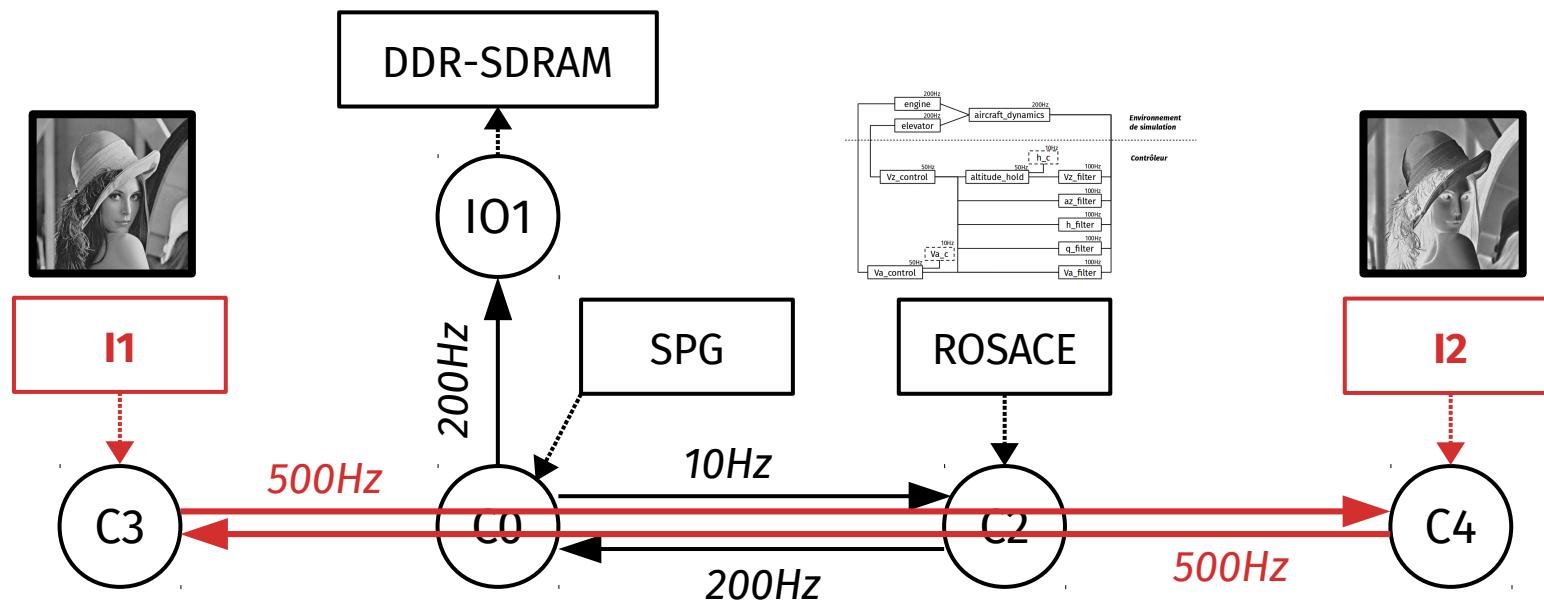
Scénario 1 : pas d'interférences



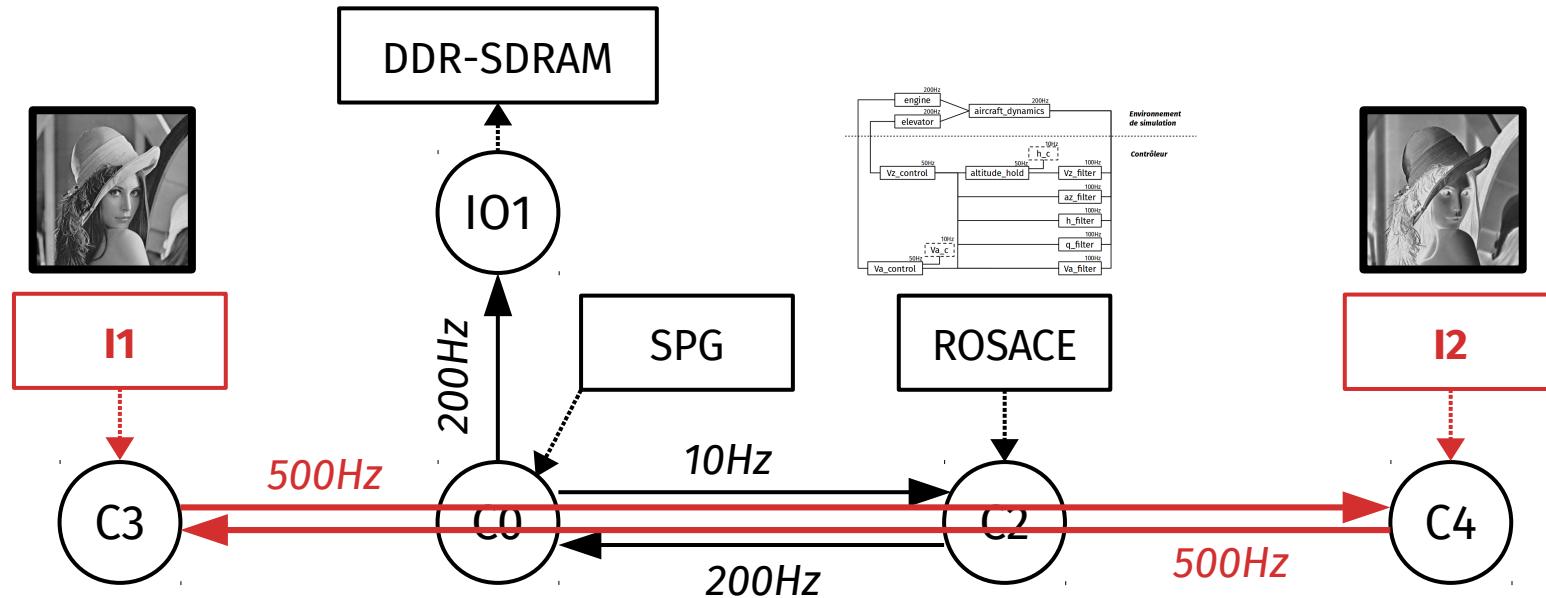
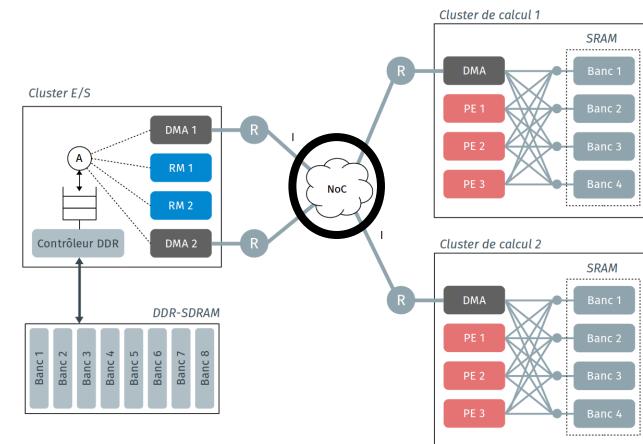
Scénario 2 : interférences sur le NoC



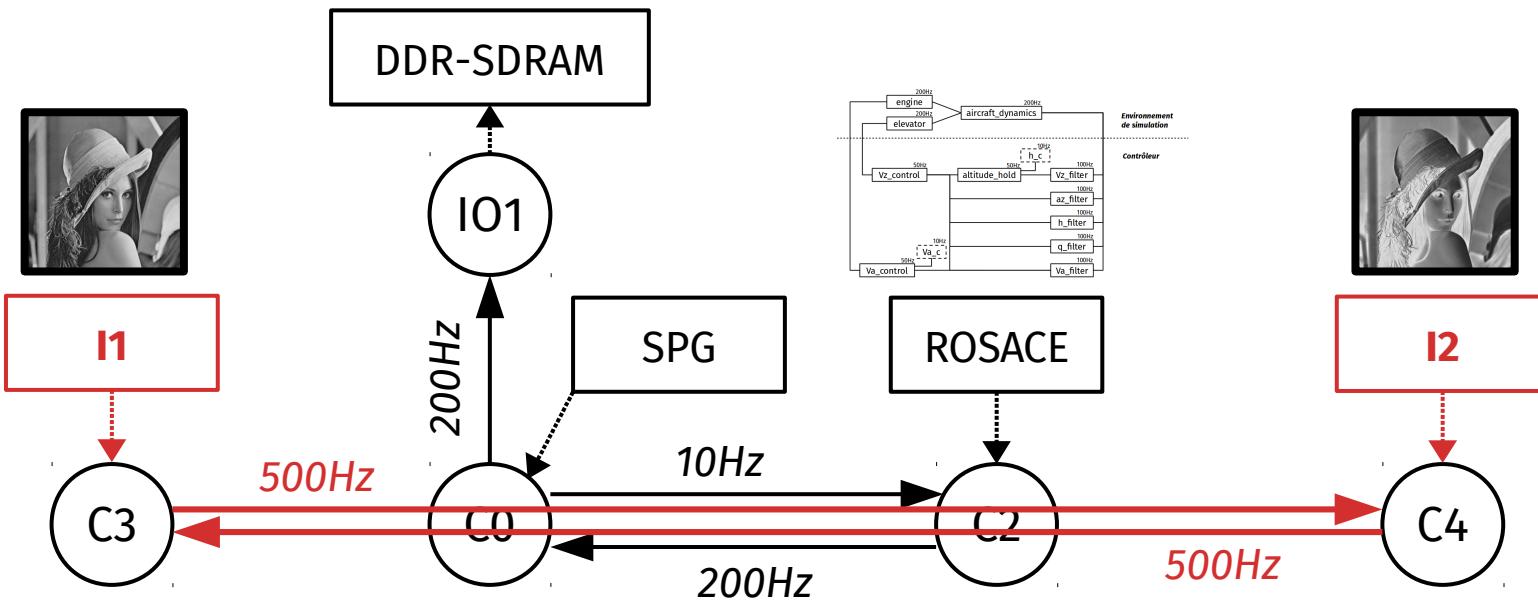
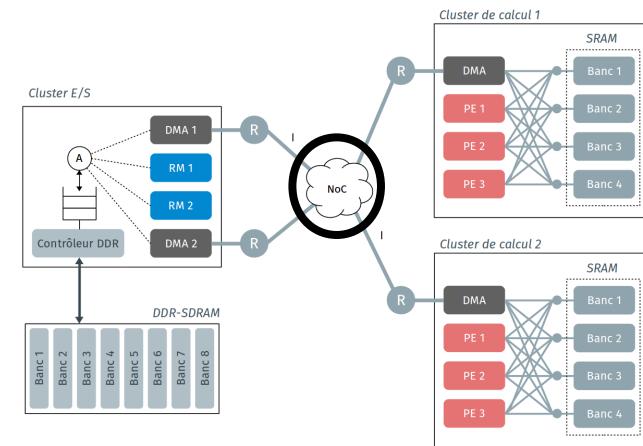
Scénario 2 : interférences sur le NoC



Scénario 2 : interférences sur le NoC

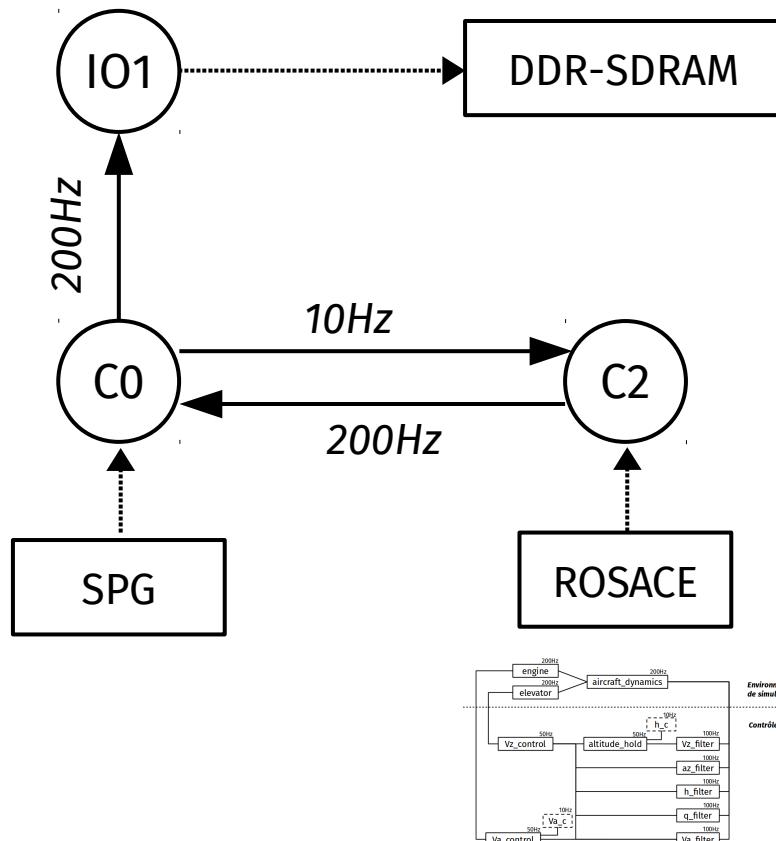


Scénario 2 : interférences sur le NoC

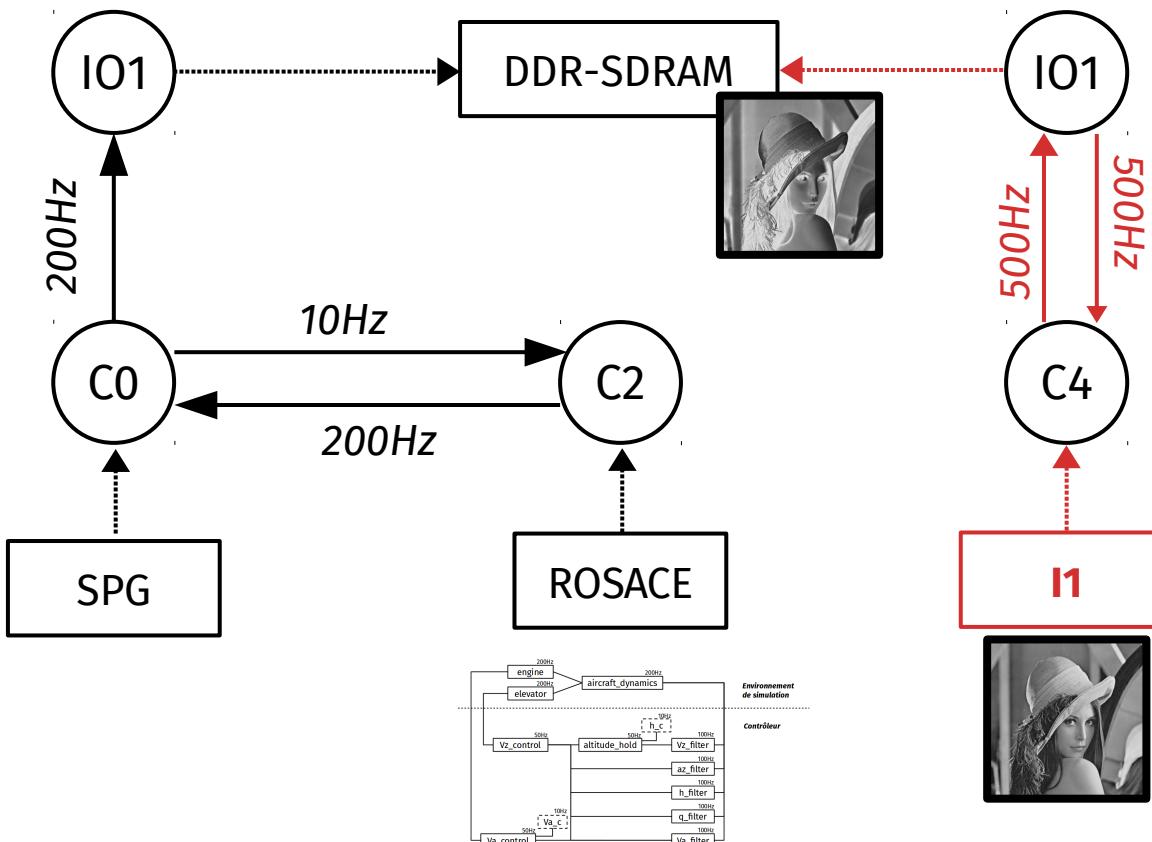


- BCET et WCET mesurés strictement **égaux**
- **100%** des paquets reçus, tous dans les **créneaux prévus**

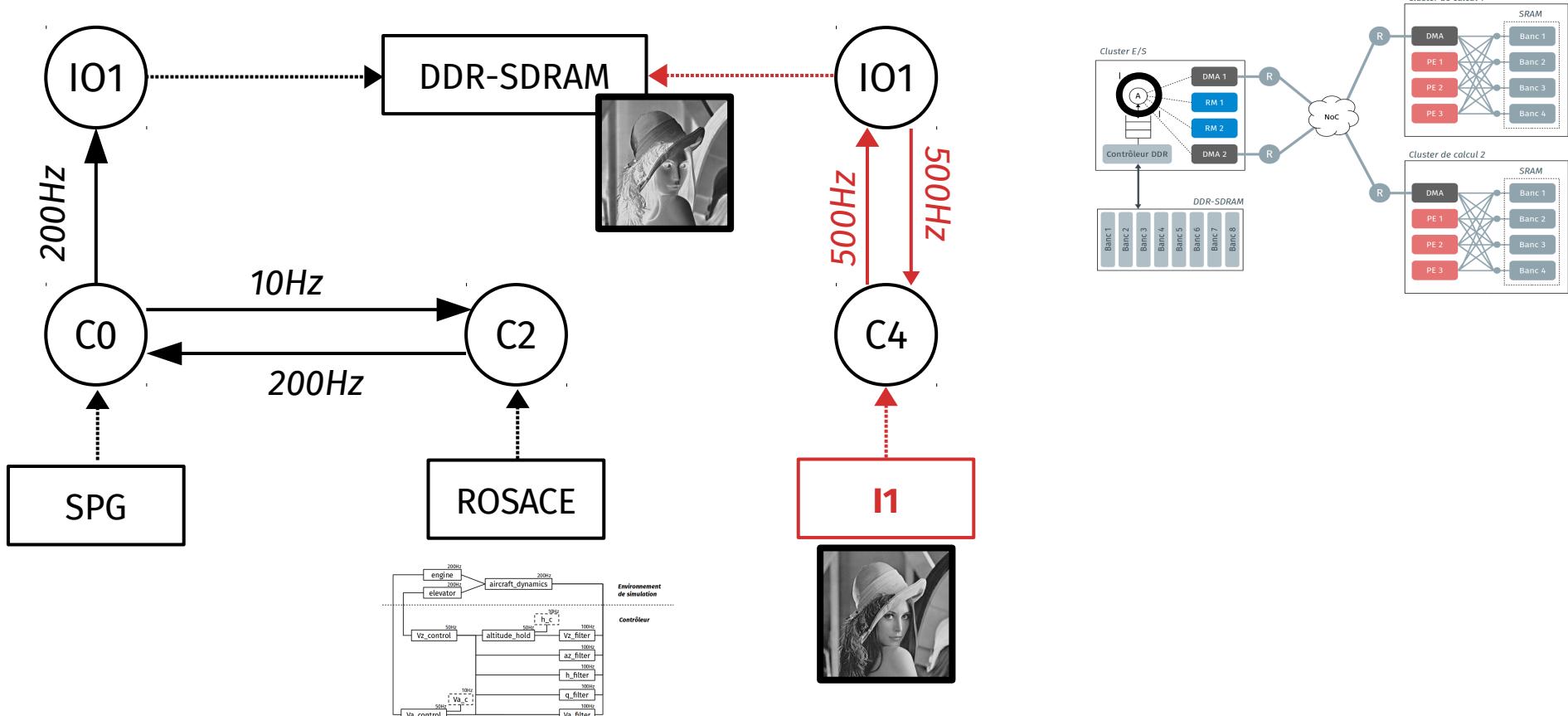
Scénario 3 : interférences sur la DDR-SDRAM



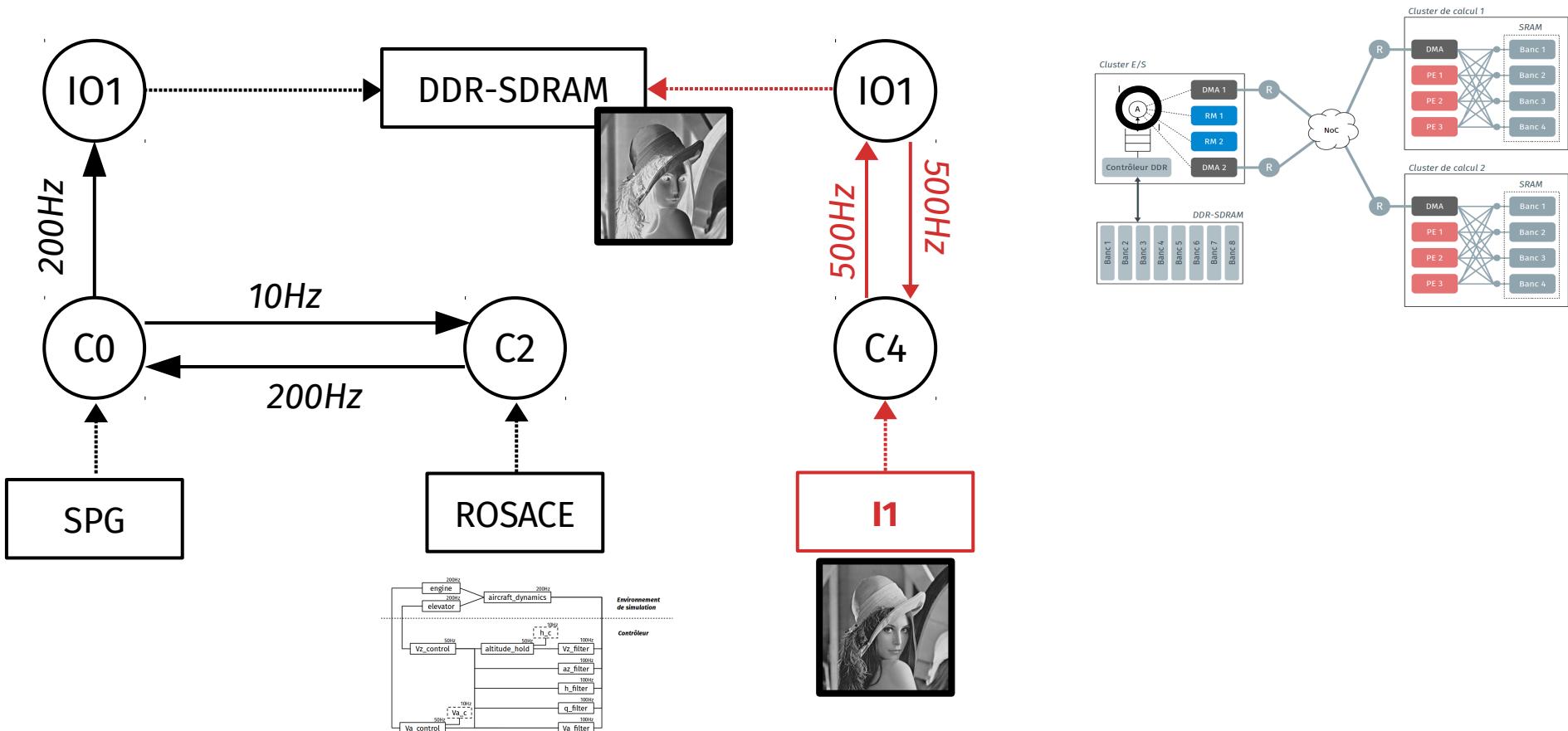
Scénario 3 : interférences sur la DDR-SDRAM



Scénario 3 : interférences sur la DDR-SDRAM



Scénario 3 : interférences sur la DDR-SDRAM



- BCET et WCET mesurés strictement **égaux**
- **100%** des paquets reçus, tous dans les **créneaux prévus**
- **Pas d'entrelacement** des transactions observé

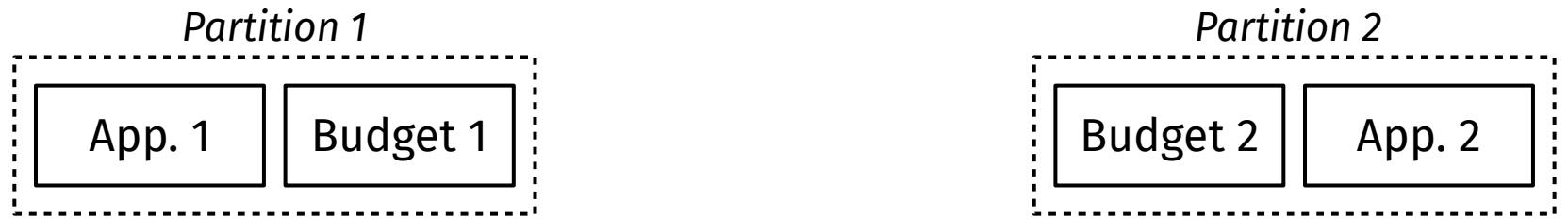
Synthèse

- ***Isolation temporelle*** sur le Kalray MPPA-256 possible avec l'utilisation d'un ***modèle d'exécution***
- Garanties du respect des règles en-ligne grâce à un ***hyperviseur***
- Résultats expérimentaux ***prometteurs*** bien que n'apportant pas une preuve formelle

Plan

1. *Kalray MPPA-256*
2. *Première contribution : Isolation temporelle*
- 3. Deuxième contribution : Placement automatique**
4. *Conclusion*

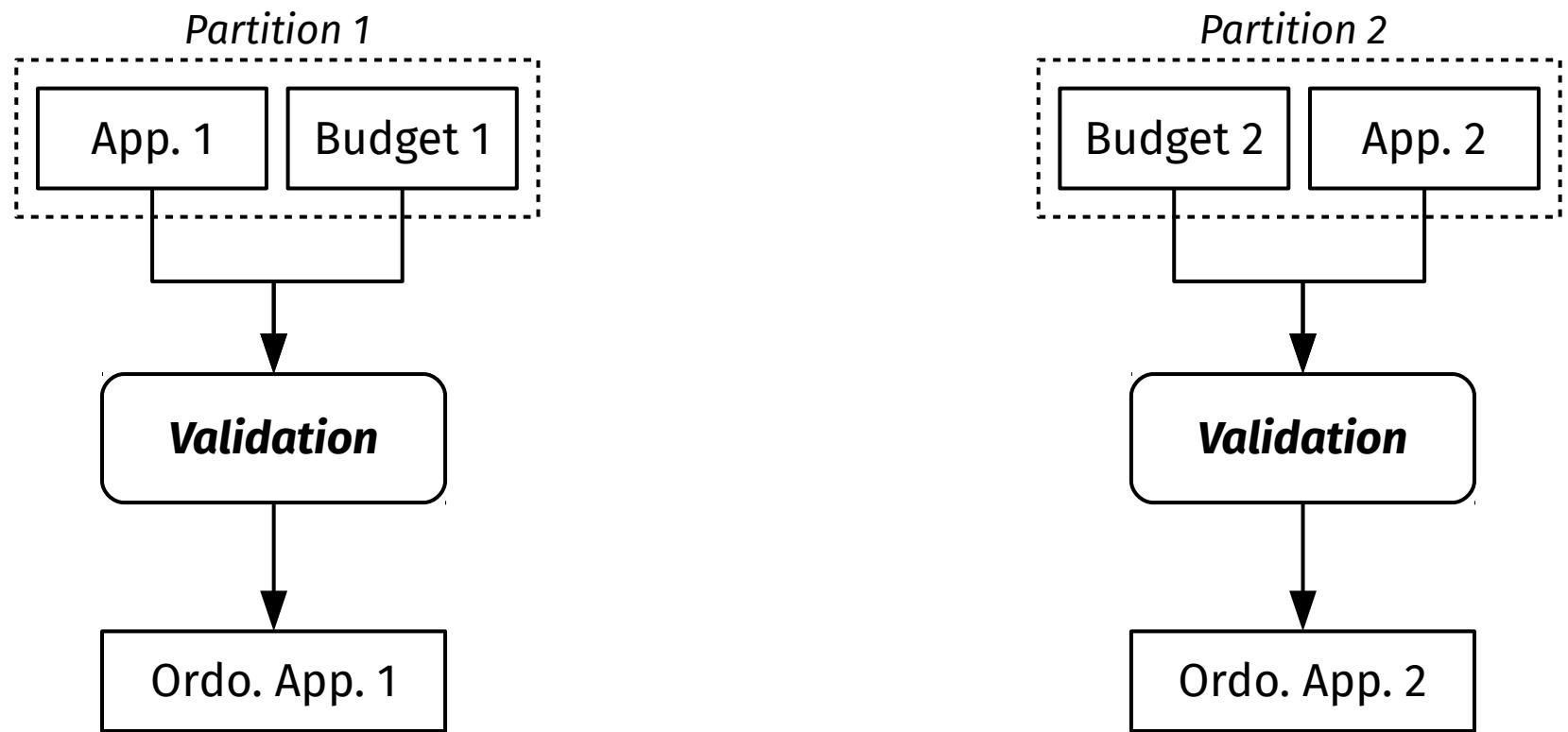
Atelier d'intégration



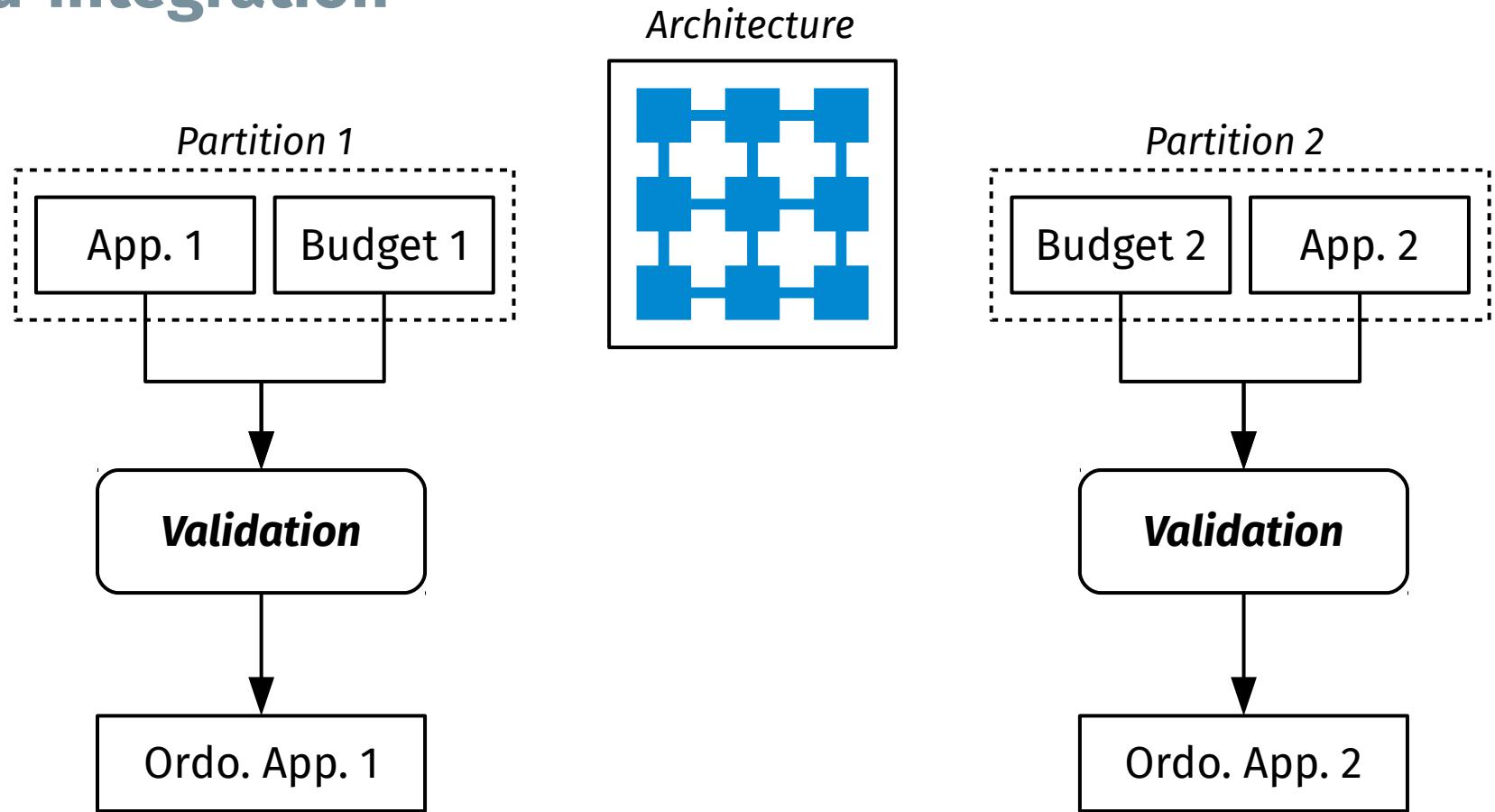
Atelier d'intégration



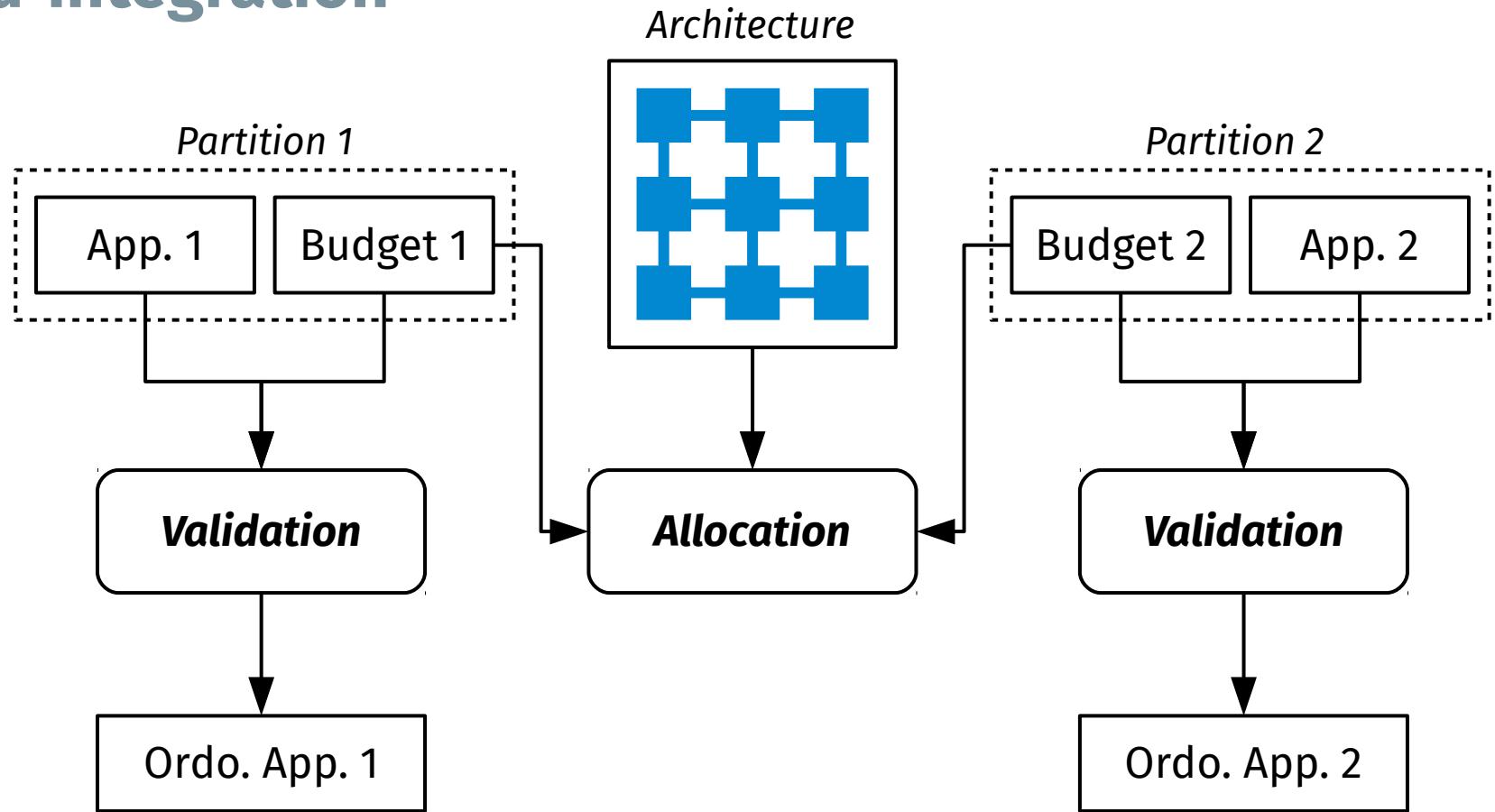
Atelier d'intégration



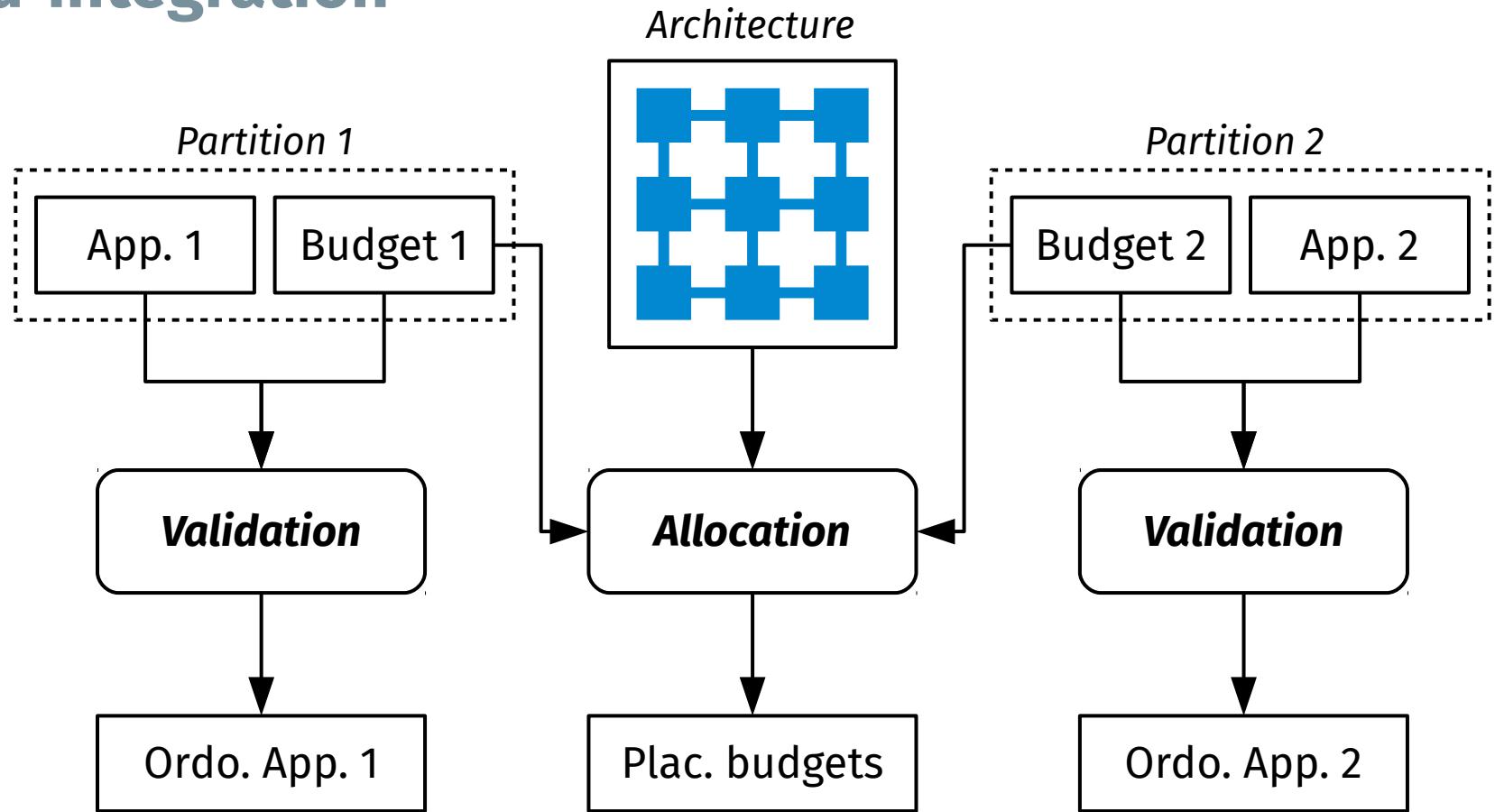
Atelier d'intégration



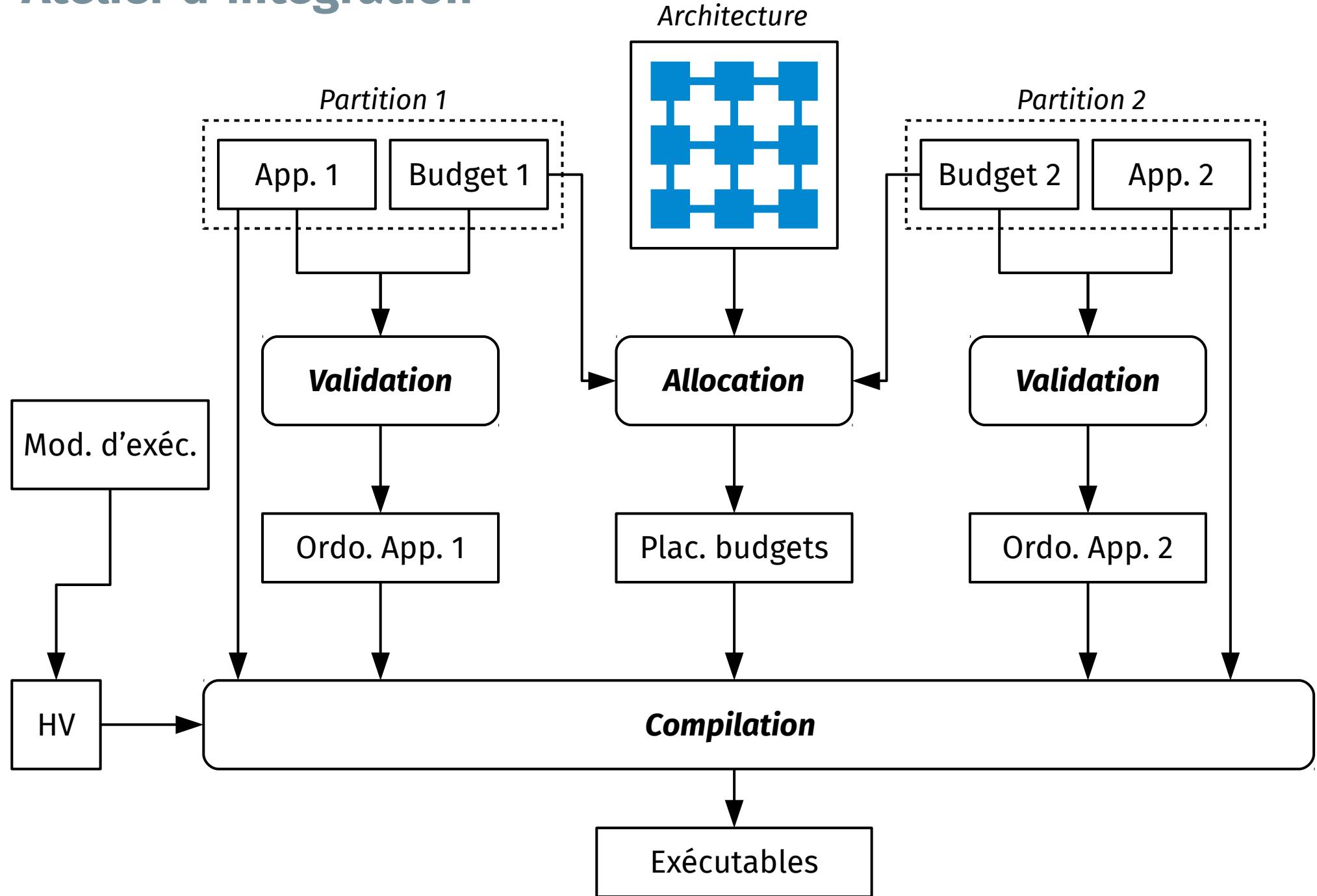
Atelier d'intégration



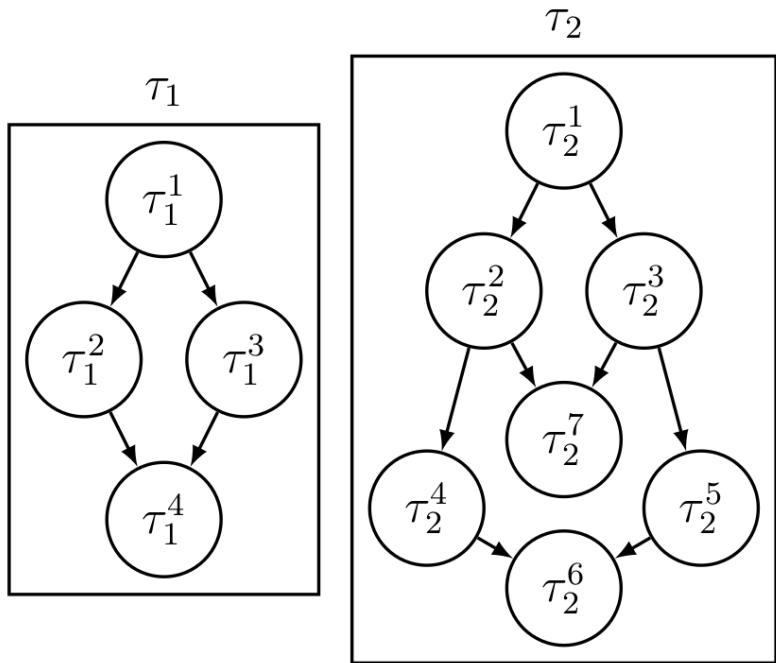
Atelier d'intégration



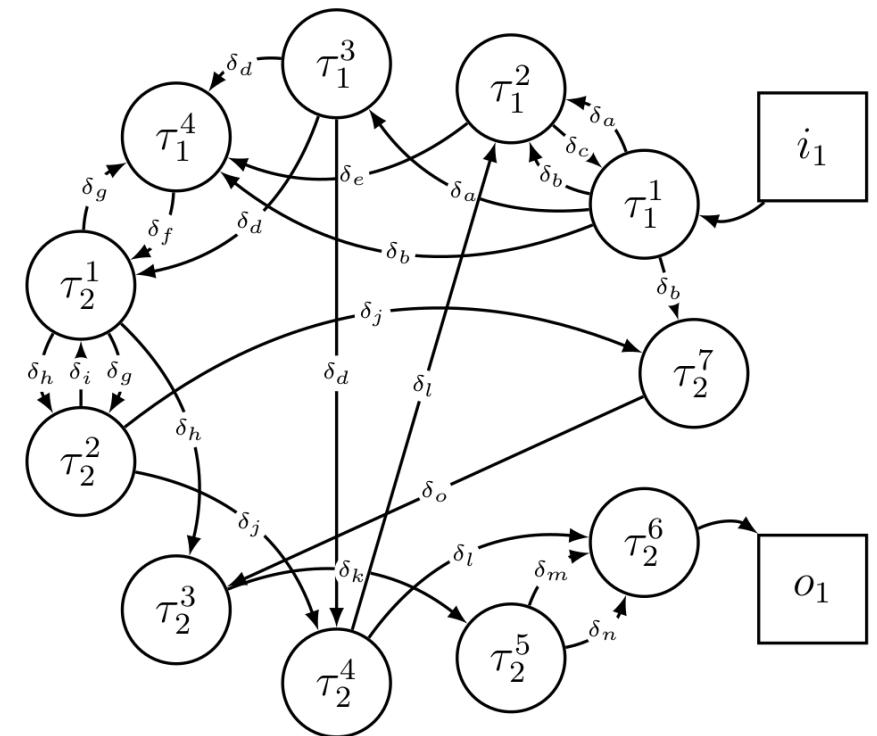
Atelier d'intégration



Modèle d'application

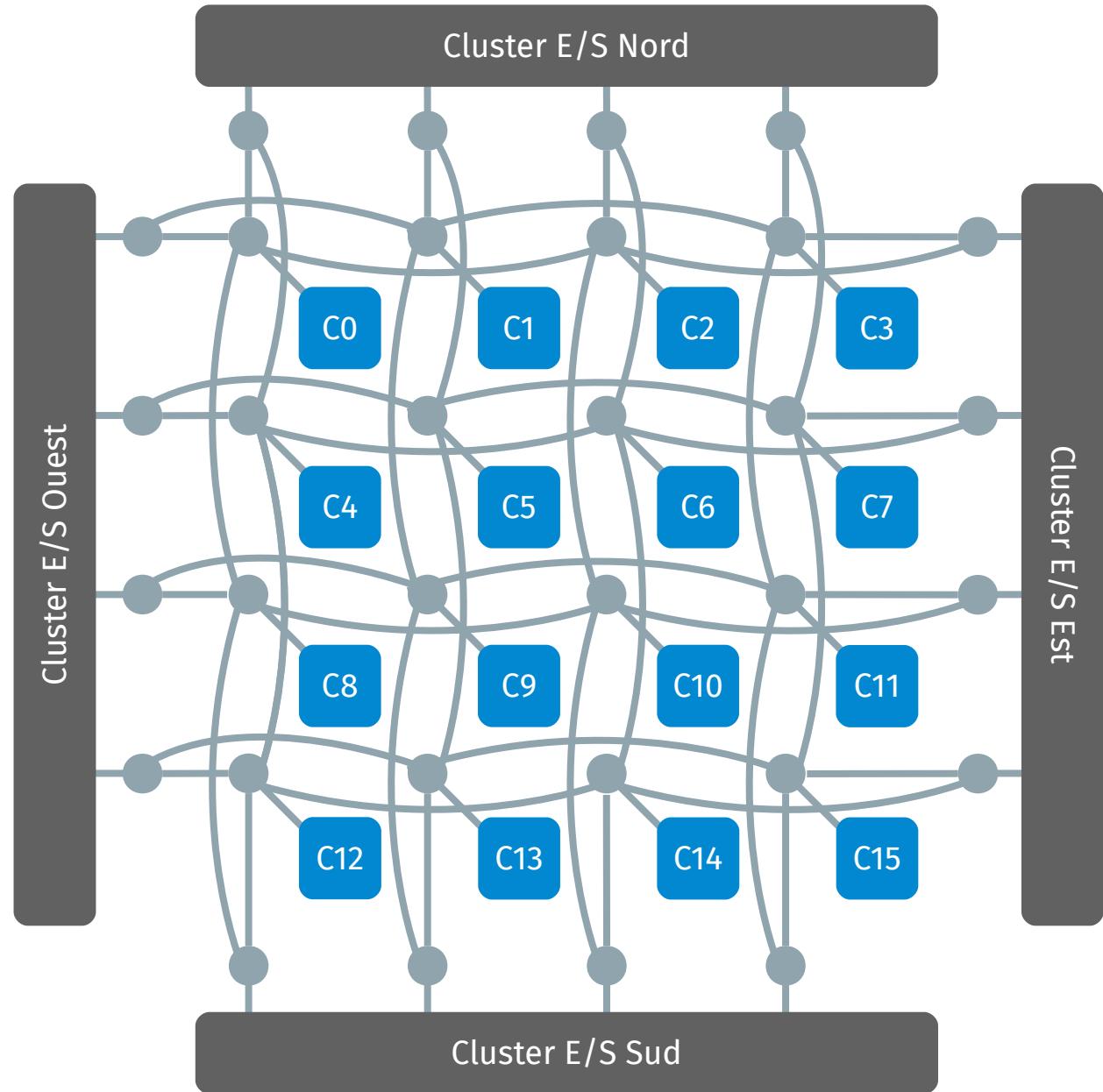


Contraintes de précédences

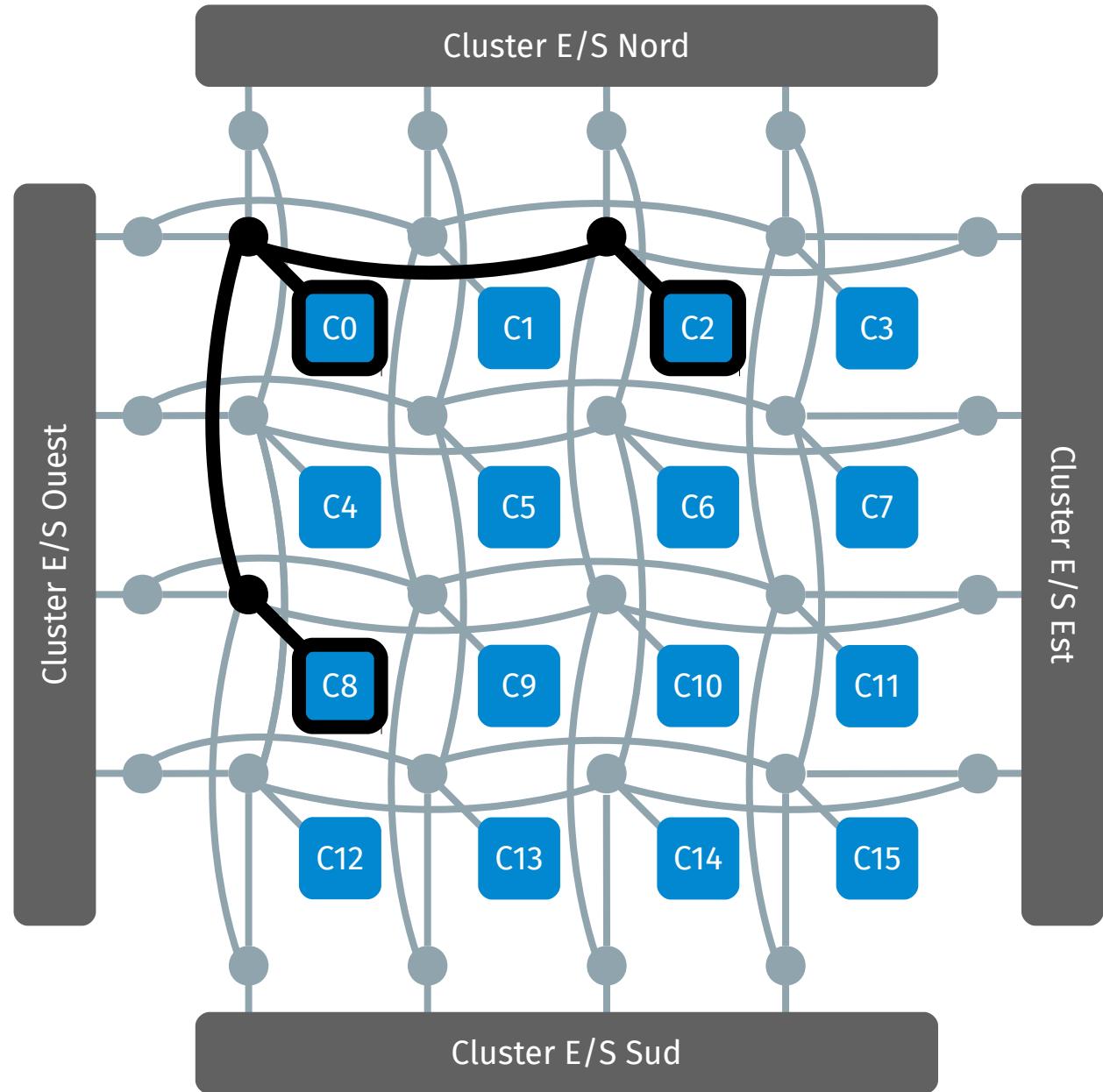


Production-consommation de données

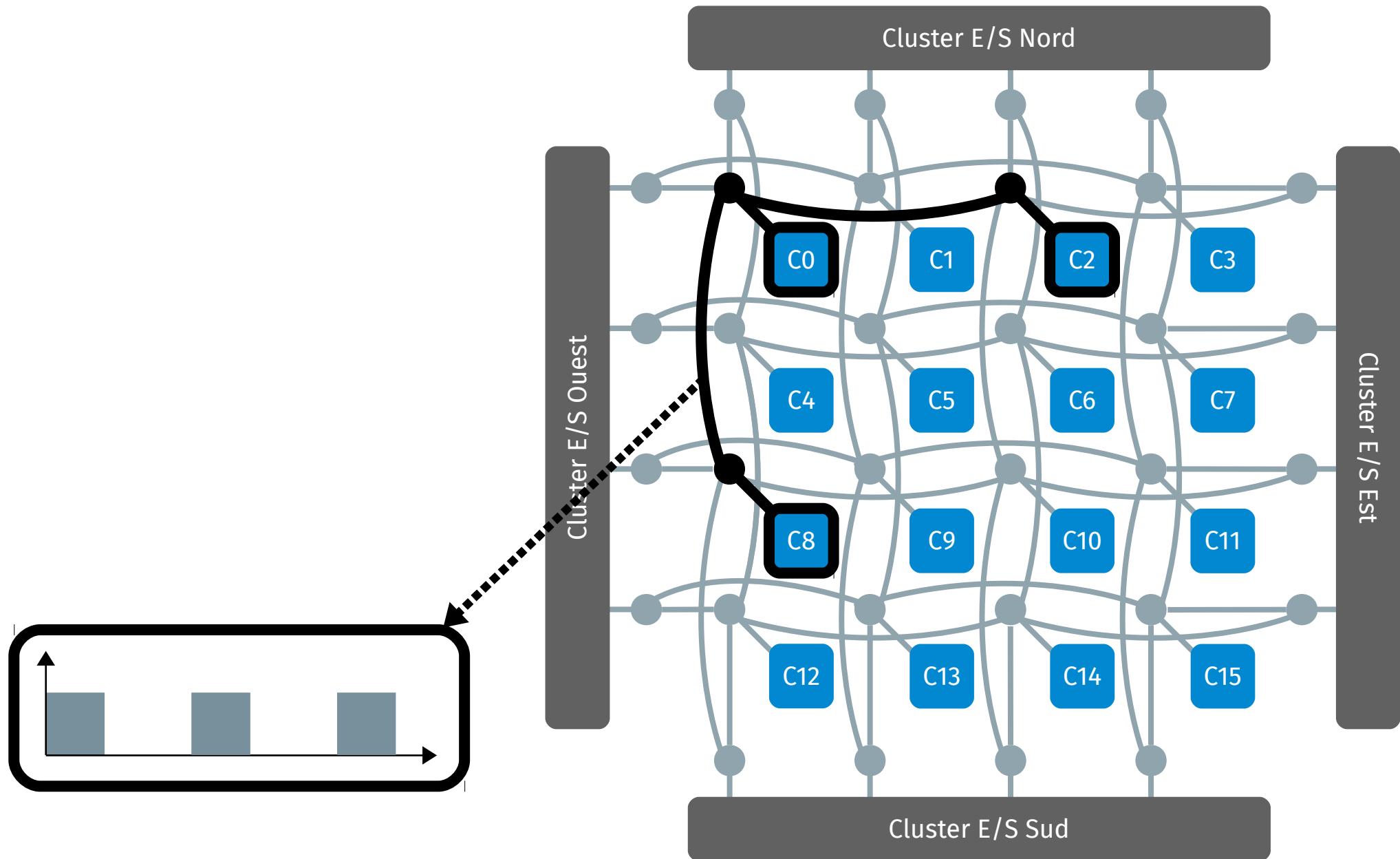
Budget



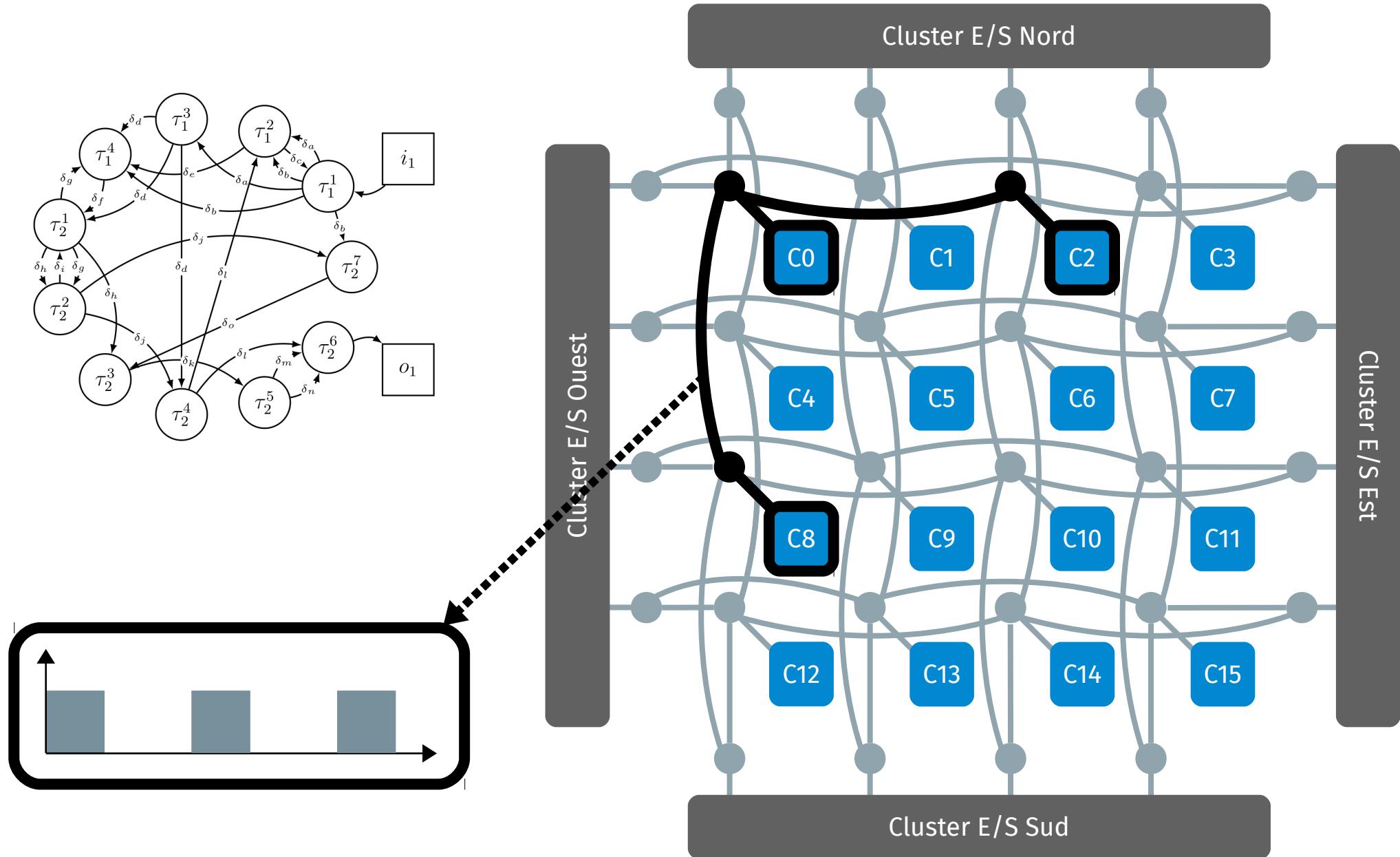
Budget



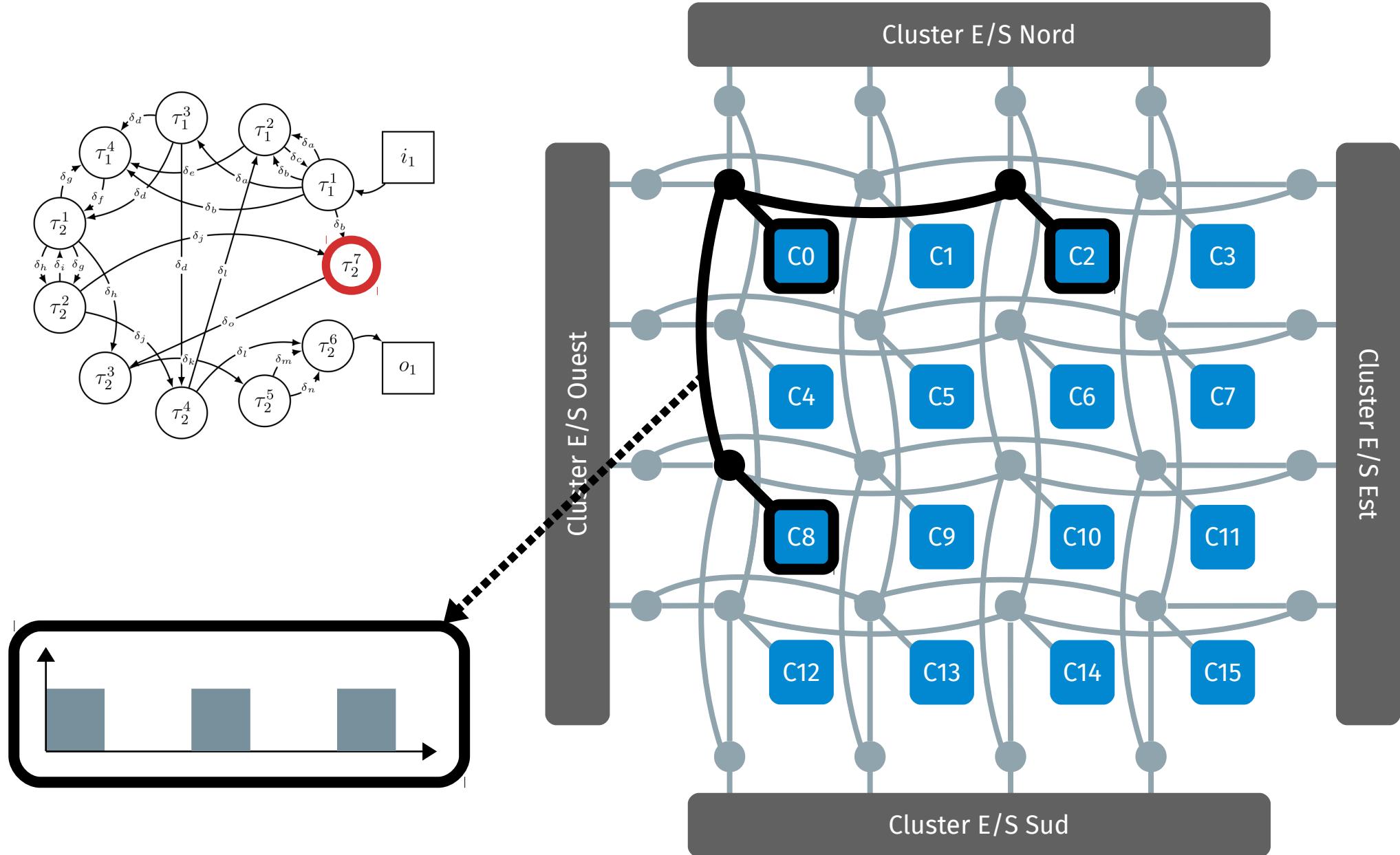
Budget



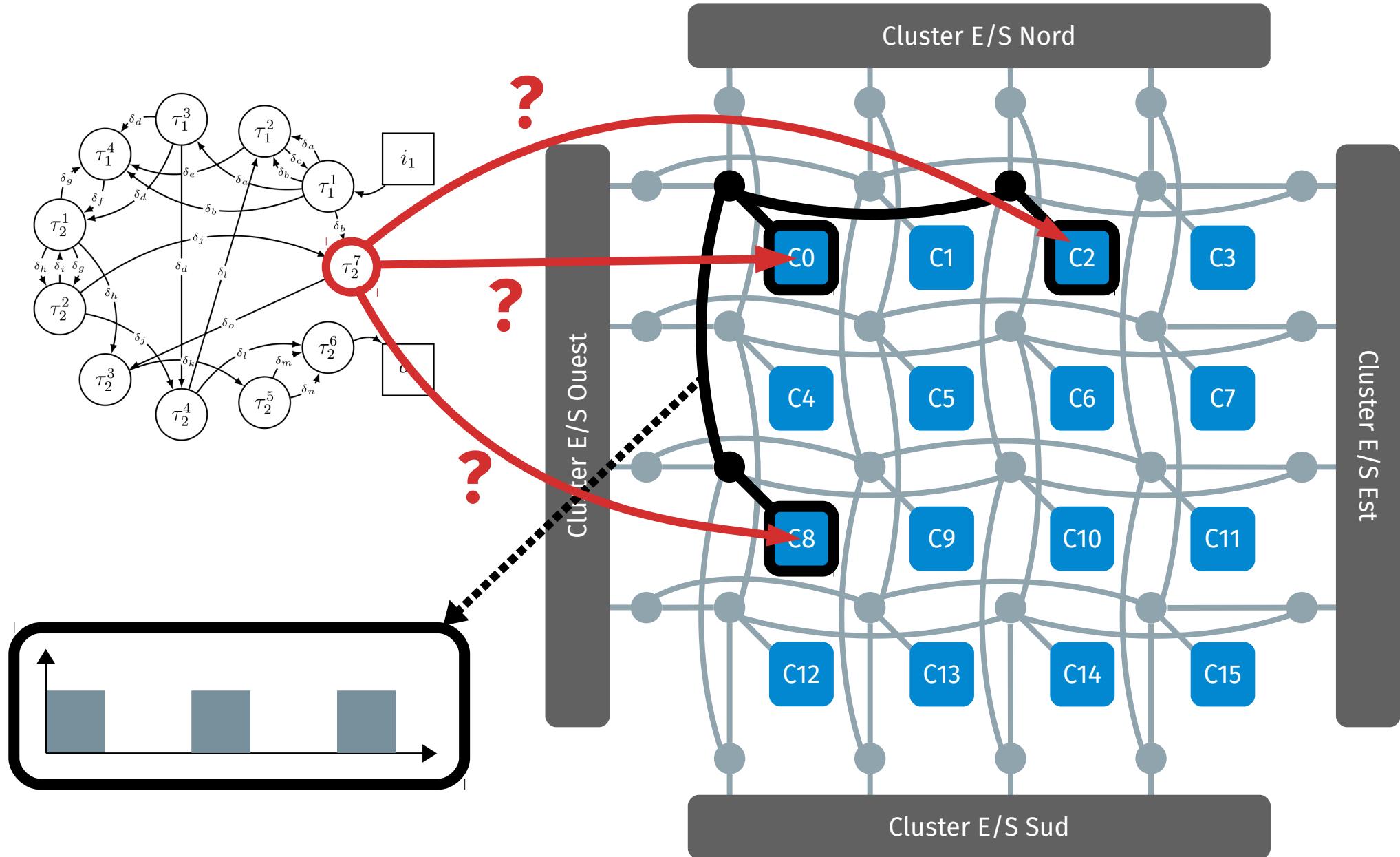
Partition



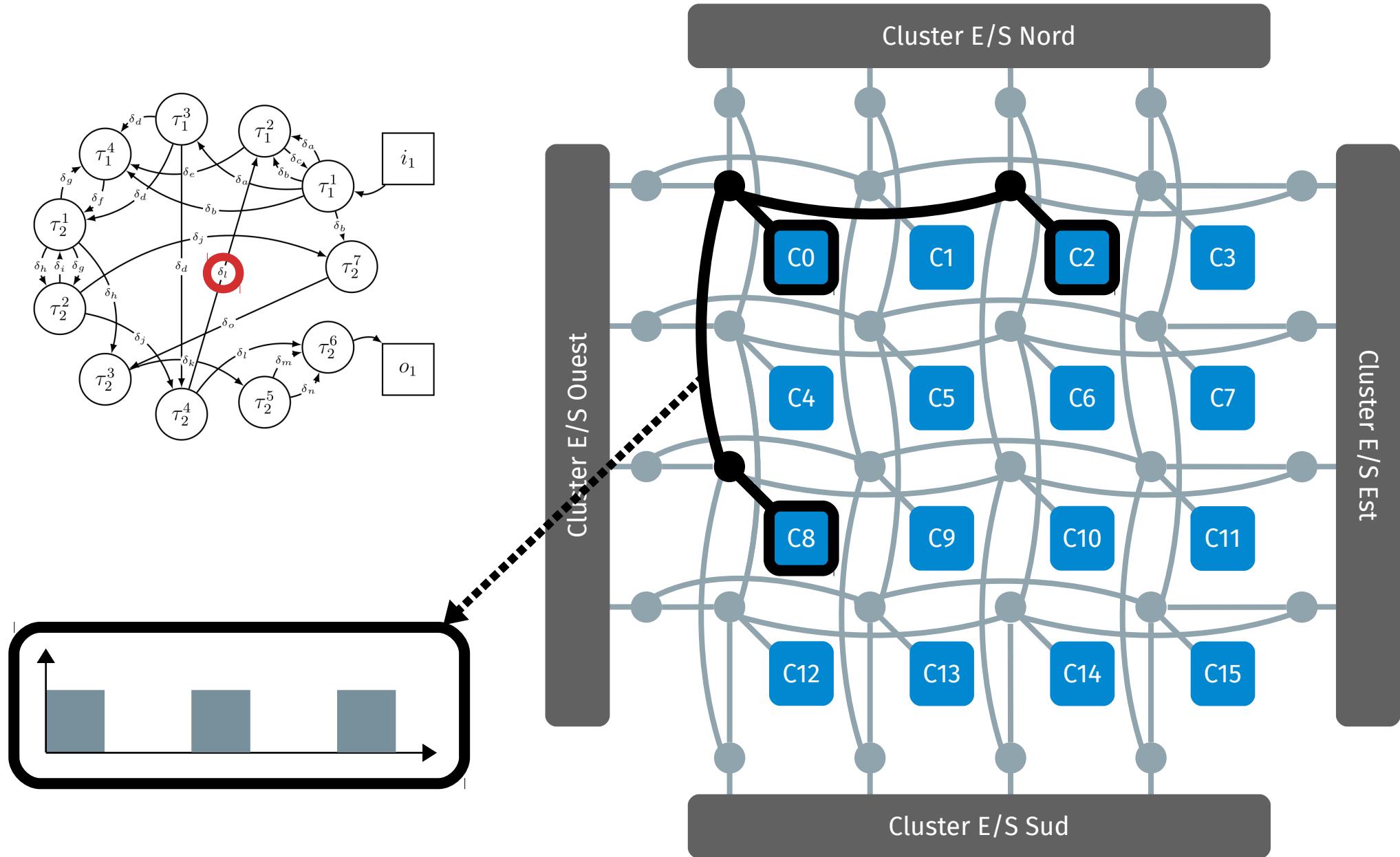
Problème



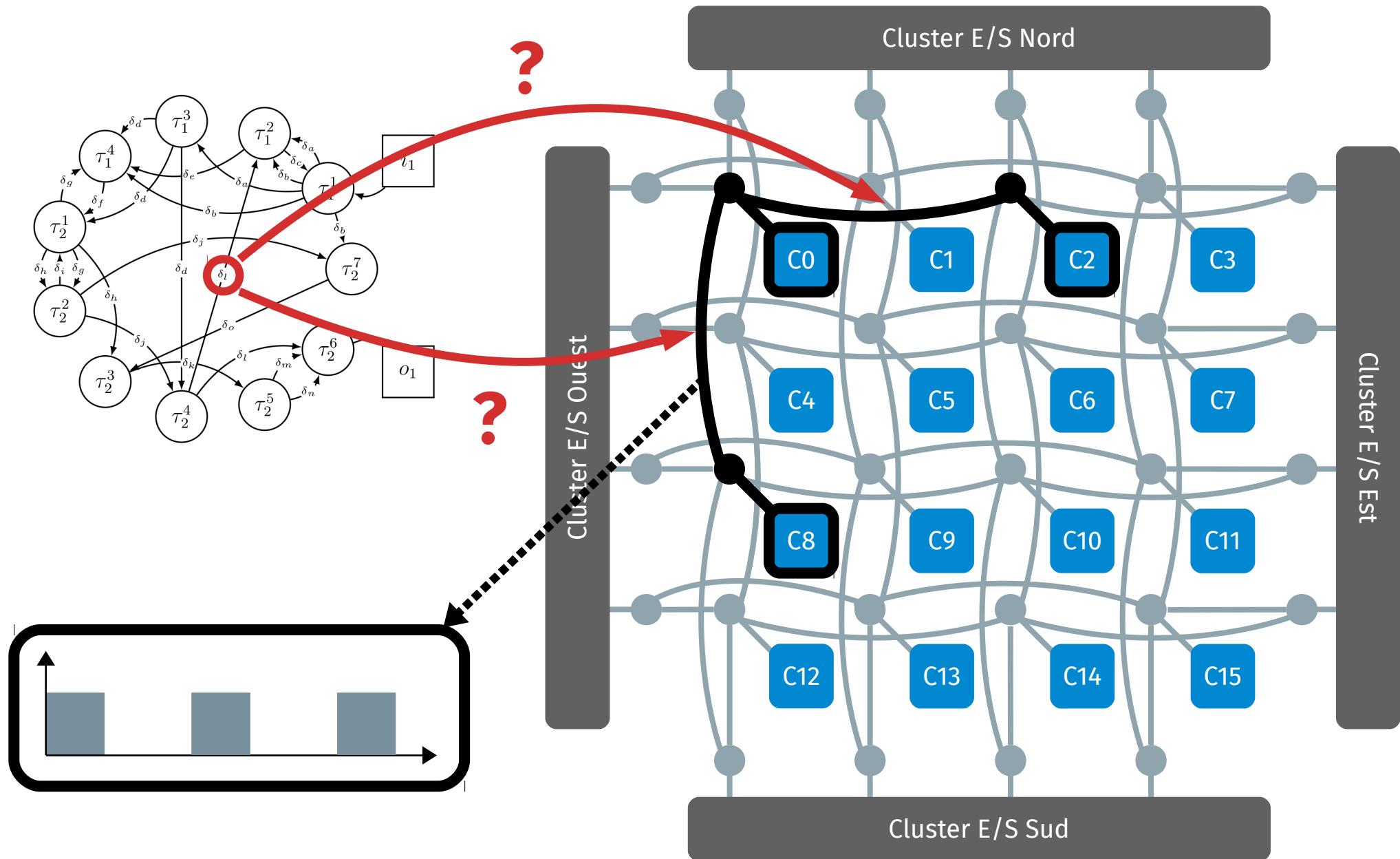
Problème



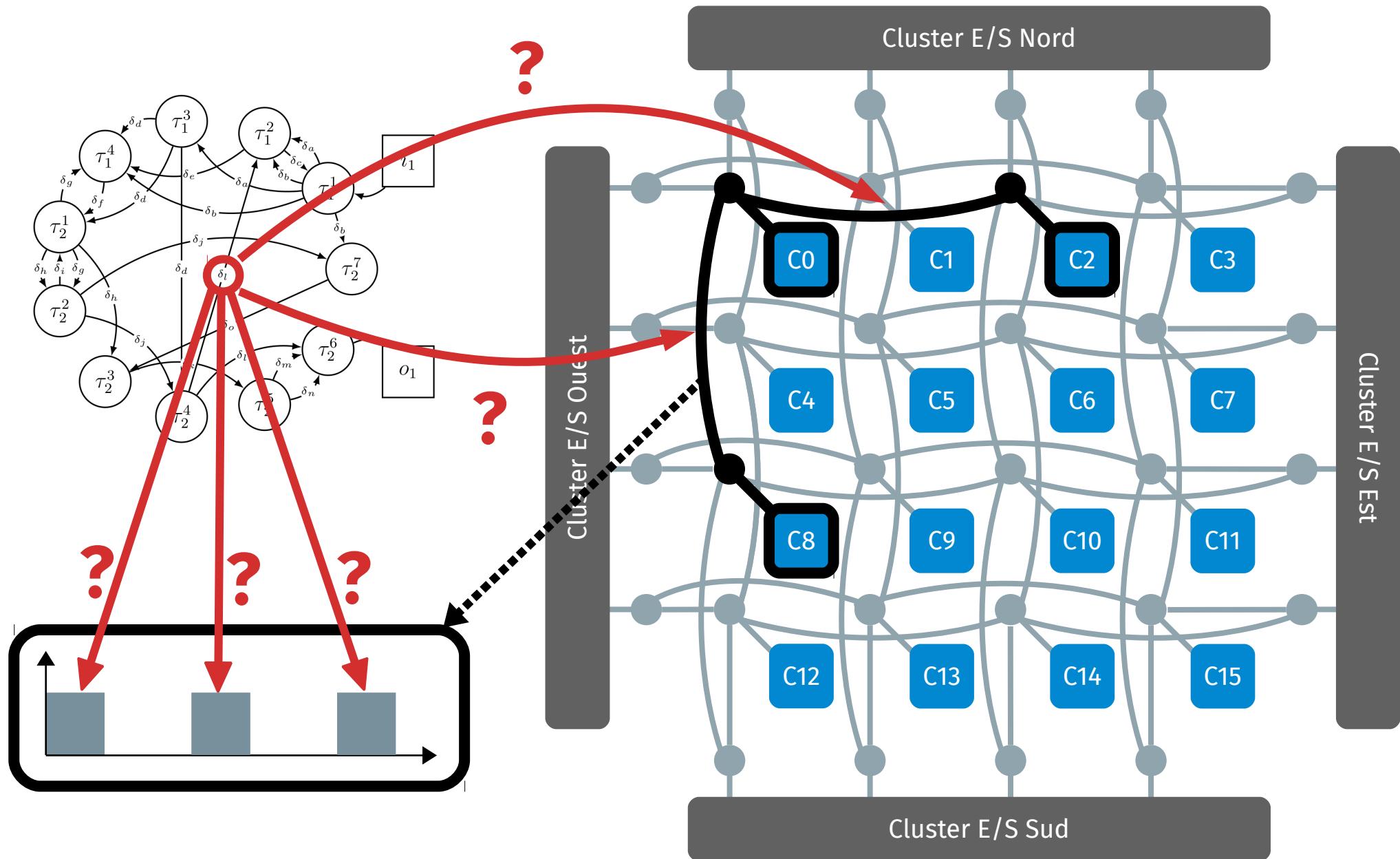
Problème

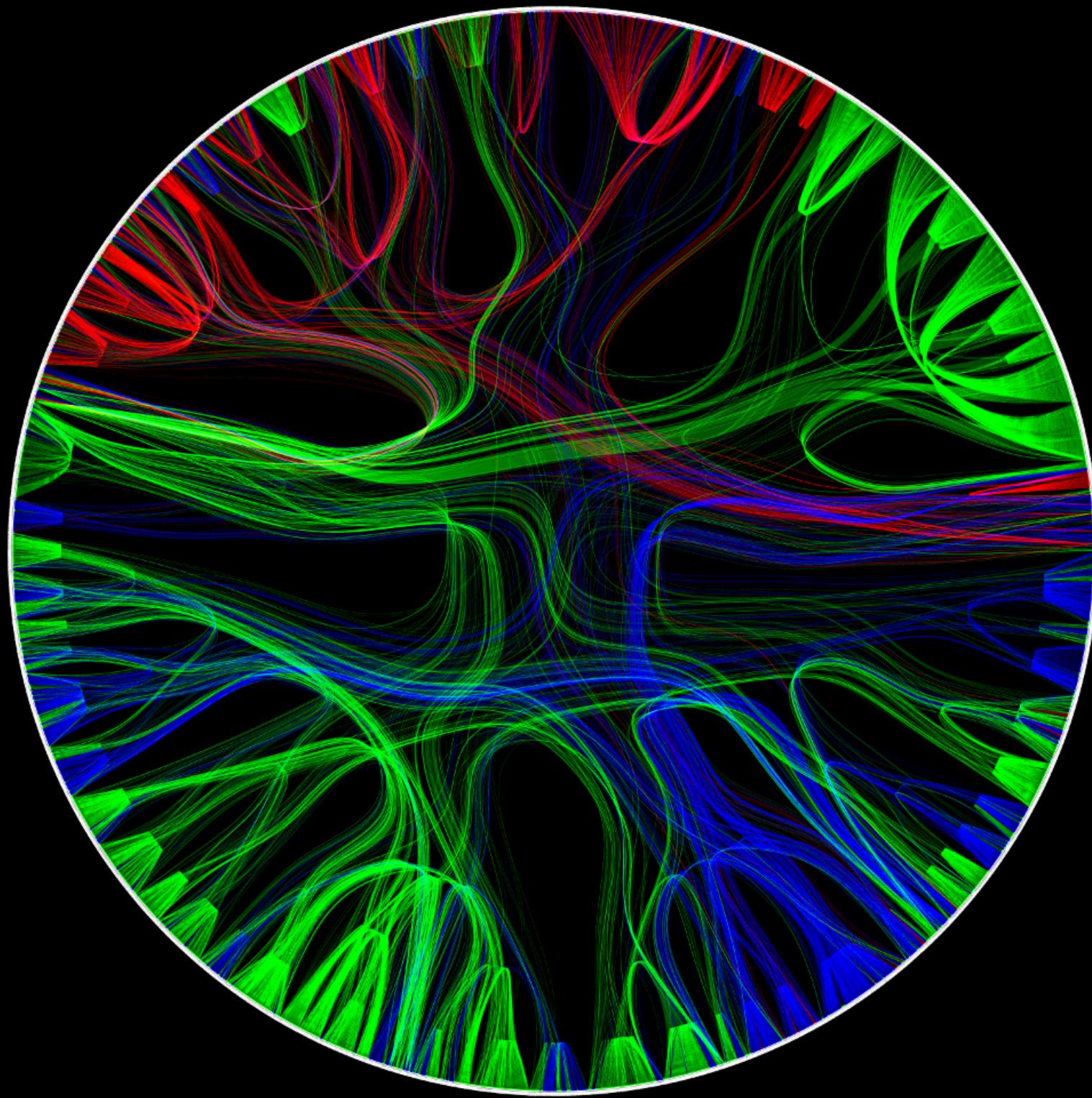


Problème



Problème





Travaux antérieurs

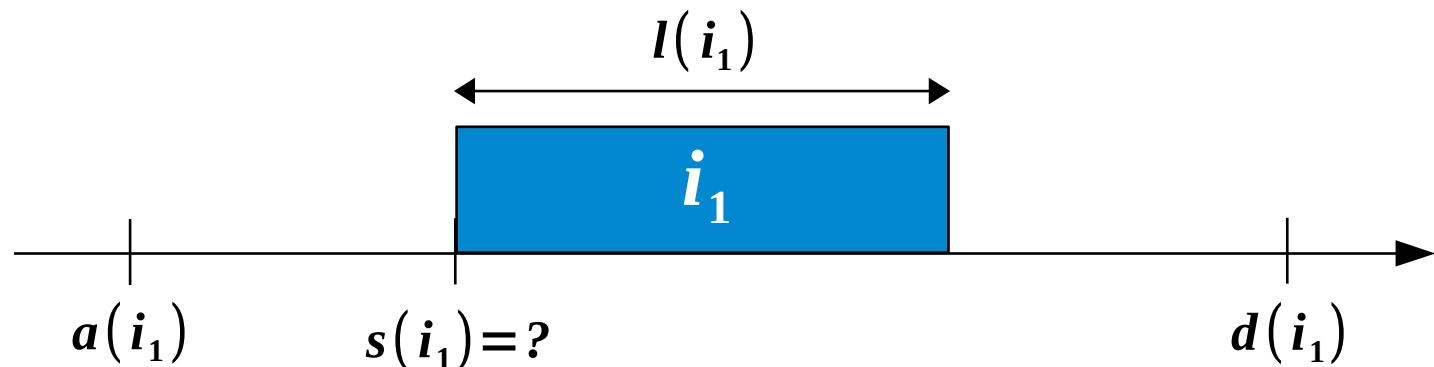
- Placement sur le **MPPA-256**
 - Flexible Time-Triggered Scheduling [8] par Giannopolou *et al.*
 - Applications AUTOSAR [9] par Becker *et al.*
- Placement sur ***d'autres pluri-cœurs***
 - LoPhT [10] par Carle *et al.*
 - Programmation par contraintes [11] par Puffisch *et al.*

Hypothèses

- Pas de récupération de code en DDR-SDRAM externe
 - Stockage dans les **mémoires locales** seulement
 - Application massive déployée sur **plusieurs clusters**
- 1 PE par cluster de calcul
 - Pression accrue sur la **gestion du NoC**
 - **Consommation énergétique** suffisamment faible
 - (*extension à plusieurs PEs possible*)

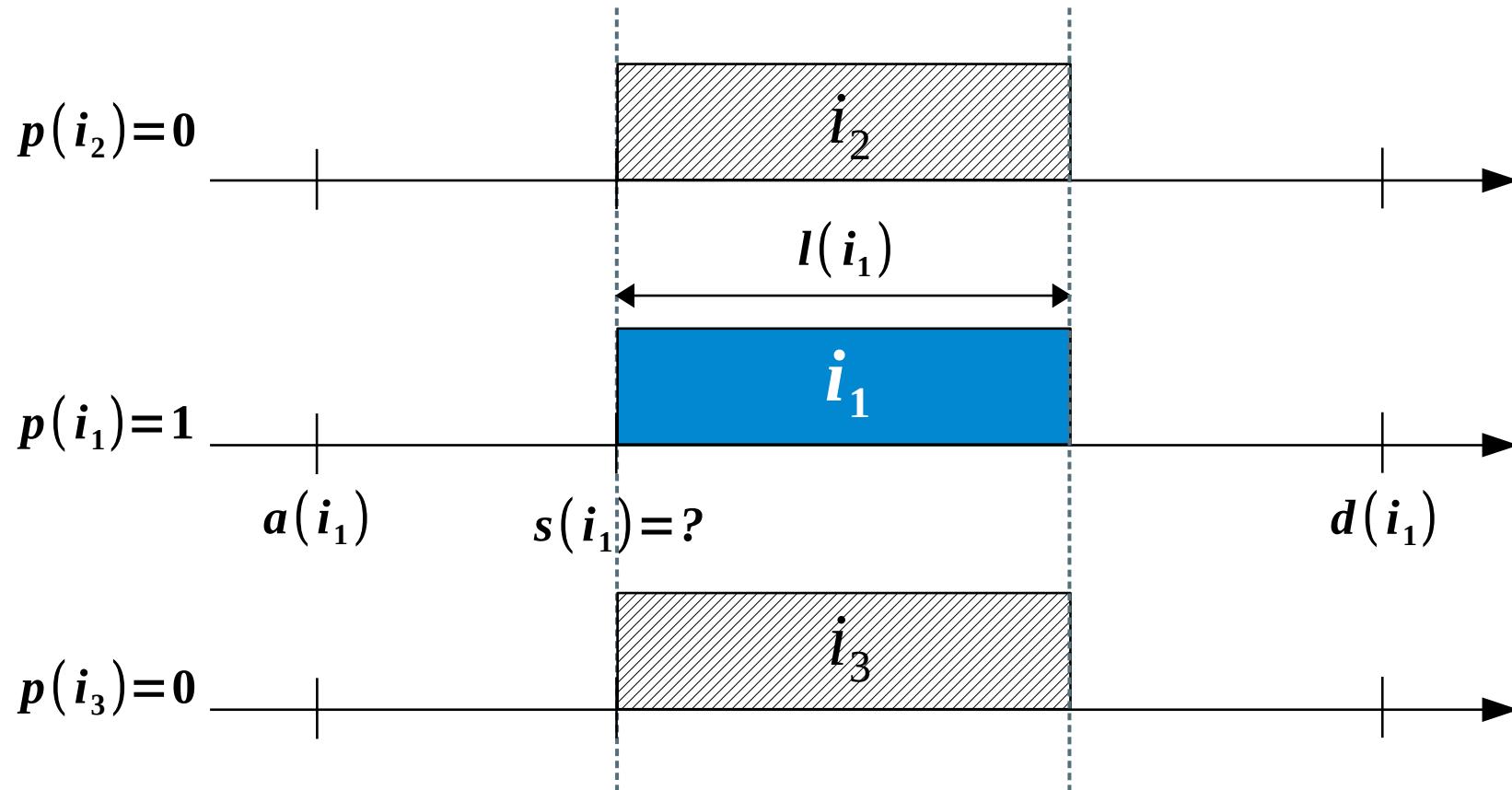
Environnement de modélisation

- Solution commune : ILP + Heuristique
- Solution proposée : ***Intervalles conditionnels***
- Introduits dans IBM ILOG CP Optimizer v2



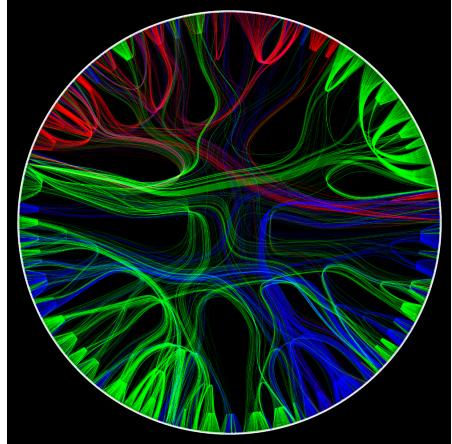
Environnement de modélisation

- Solution commune : ILP + Heuristique
- Solution proposée : **Intervalles conditionnels**
- Introduits dans IBM ILOG CP Optimizer v2

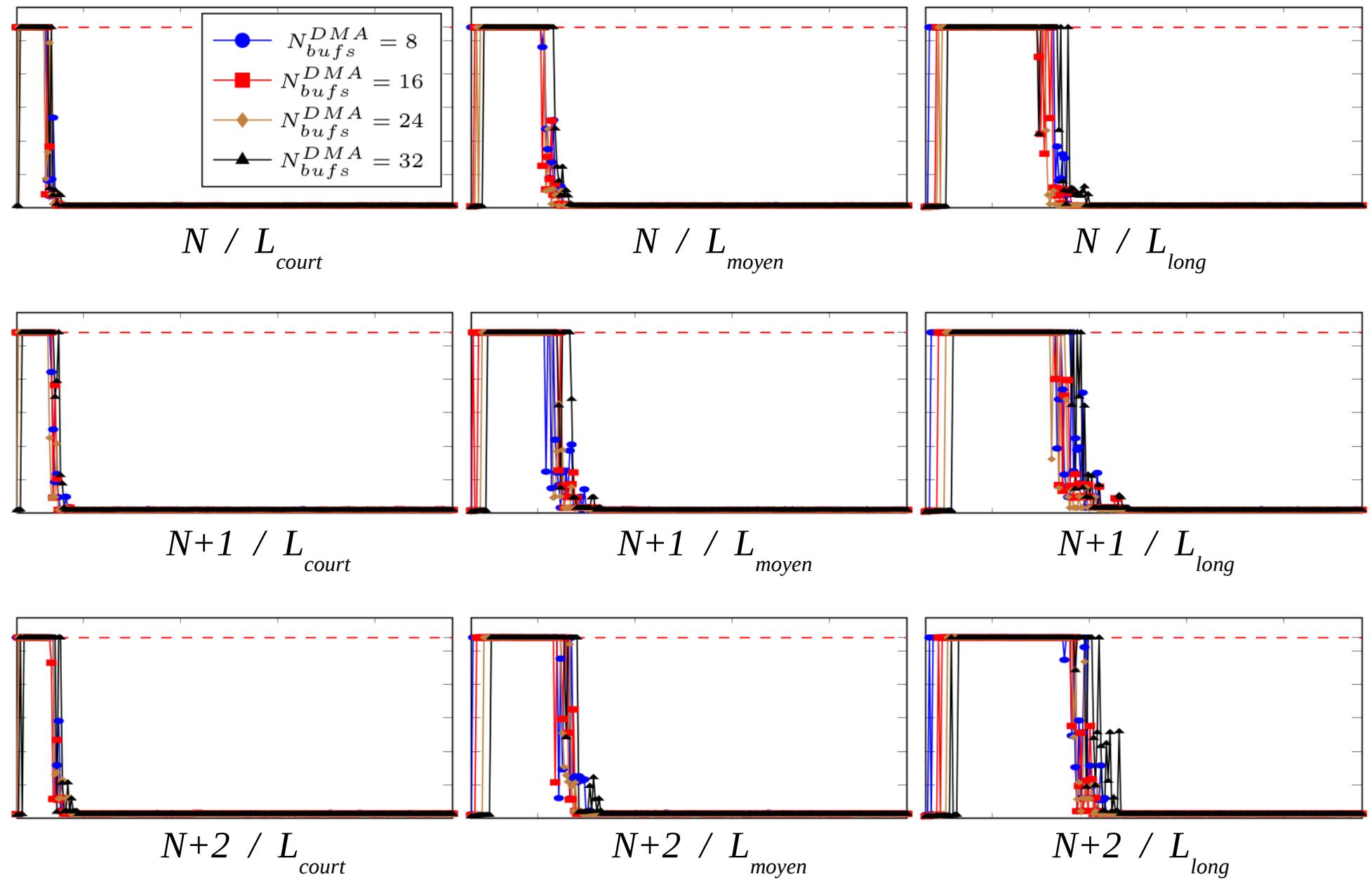


Validation expérimentale

- Dimension du problème
 - ~ **100 000 instances** de sous-tâches et données
 - Des **millions** de contraintes et variables de décision
- Objectifs
 - Réduction de l'hyperpériode (maximisation de l'**utilisation**)
 - Modification du **budget**
 - Nombre de clusters de calcul $\in \{ N ; N+1 ; N+2 \}$
 - Longueur des créneaux de communication $\in \{ L_{court} ; L_{moyen} ; L_{long} \}$
 - Configuration de l'**hyperviseur**
 - Autonomie du DMA $\in [8 , 32]$



Résultats



Synthèse

- La parallélisation d'applications de **grandes tailles** est possible avec la **programmation par contraintes**
- La formalisation par **intervalles** permet une résolution **rapide et efficace**
- L'étude paramétrique semble indiquer que la **gestion du NoC** reste limitante pour plus de performances

Plan

1. *Kalray MPPA-256*
2. *Première contribution : Isolation temporelle*
3. *Deuxième contribution : Placement automatique*
- 4. Conclusion**

Conclusion

- Les **pluri-cœurs** sont de **bons candidats** pour la conception de calculateurs avioniques performants
- L'**isolation temporelle** est possible grâce un **modèle d'exécution** supporté par un **hyperviseur**
- Le placement **automatique et efficace** d'applications de **grandes tailles** est possible grâce à une **formalisation adaptée**

Perspectives

- Optimisations du ***positionnement des données*** en mémoire
- Automatisation de ***l'allocation*** des budgets
- Automatisation du ***provisionnement*** des budgets
- ***Assouplissement*** du modèle d'exécution et de l'hyperviseur
- Évaluation ***d'autres budgets*** avec d'autres choix d'implémentation
- Prise en compte d'applications de ***types différents***

MERCI.

Avez-vous des questions ?

Références

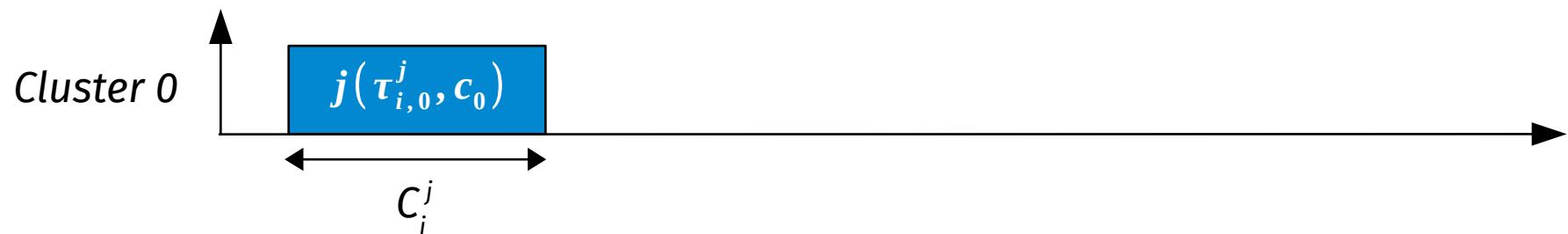
- [1] Stuart Fisher. *Certifying Applications in a Multi-Core Environment: The World's First Multi-Core Certification to SIL 4*. 2014
- [2] Xavier Jean, Daniel Faura, Marc Gatti, Laurent Pautet, and Thomas Robert. "Ensuring robust partitioning in multicore platforms for IMA systems". In: *31st IEEE/AIAA Digital Avionics Systems Conference (DASC'12)*. 2012
- [3] Heechul Yun, Gang Yao, R. Pellizzoni, M. Caccamo, and Lui Sha. "MemGuard: Memory bandwidth reservation system for efficient performance isolation in multi-core platforms". In: *19th Real-Time and Embedded Technology and Applications Symposium (RTAS'13)*. 2013, pp. 55–64
- [4] Rodolfo Pellizzoni, Emiliano Betti, Stanley Bak, Gang Yao, John Criswell, Marco Caccamo, and Russell Kegley. "A Predictable Execution Model for COTS-based Embedded Systems". In: *17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'11)*. 2011, pp. 269–279
- [5] Guy Durrieu, Madeleine Faugère, Sylvain Girbal, Daniel Gracia Pérez, Claire Pagetti and Wolfgang Puffitsch. "Predictable Flight Management System Implementation on a Multicore Processor". In: *7th Conference on Embedded Real Time Software and Systems (ERTS'14)*. 2014
- [6] Rohan Tabish, Reanto Mancuso, Saud Wasly, Ahmed Alhammad, Sujit S. Phatak, Rodolfo Pellizzoni, and Marco Caccamo. "A Real-Time Scratchpad-Centric OS for Multi-Core Embedded Systems". In: *22nd IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'16)*. 2016, pp. 1–11
- [7] Claire Pagetti, David Saussié, Romain Gratia, Eric Noulard and Pierre Siron "The ROSACE case study: From Simulink specification to multi/many-core execution". In: *20th Real-Time and Embedded Technology and Applications Symposium (RTAS'14)*. 2014
- [8] Georgia Giannopoulou, Nikolay Stoimenov, Pengcheng Huang, Lothar Thiele, and Benoît Dupont de Dinechin. "Mixed-criticality scheduling on cluster-based manycores with shared communication and storage resources". In: *Real-Time Systems* (2015)
- [9] Matthias Becker, Dakshina Dasari, Borislav Nicolic, Benny Åkesson, Vincent Nélis, and Thomas Nolte. "Contention-Free Execution of Automotive Applications on a Clustered Many-Core Platform". In: *28th Euromicro Conference on Real-Time Systems (ECRTS'16)*. 2016
- [10] Thomas Carle, Manel Djemal, Dumitru Potop-Butucaru, and Robert De Simone. "Static mapping of real-time applications onto massively parallel processor arrays". In: *14th International Conference on Application of Concurrency to System Design (ACSD'14)*. 2014, pp. 112–121
- [11] Wolfgang Puffitsch, Éric Noulard, and Claire Pagetti. "Off-line mapping of multi-rate dependent task sets to many-core platforms". In: *Real-Time Systems* 51.5 (2015), pp. 526–565
- [12] Silviu S. Craciunas and Ramon Serna Oliver. "Combined task- and network-level scheduling for distributed time-triggered systems". In: *Real-Time Systems* 52.2 (2016), pp. 161–200

Niveau d'ordonnancement

- Un intervalle représente une ***instance*** de sous-tâche

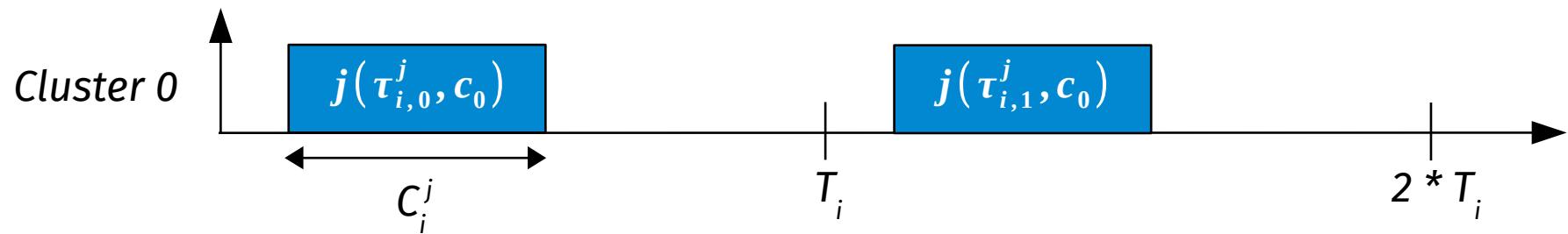
Niveau d'ordonnancement

- Un intervalle représente une ***instance*** de sous-tâche



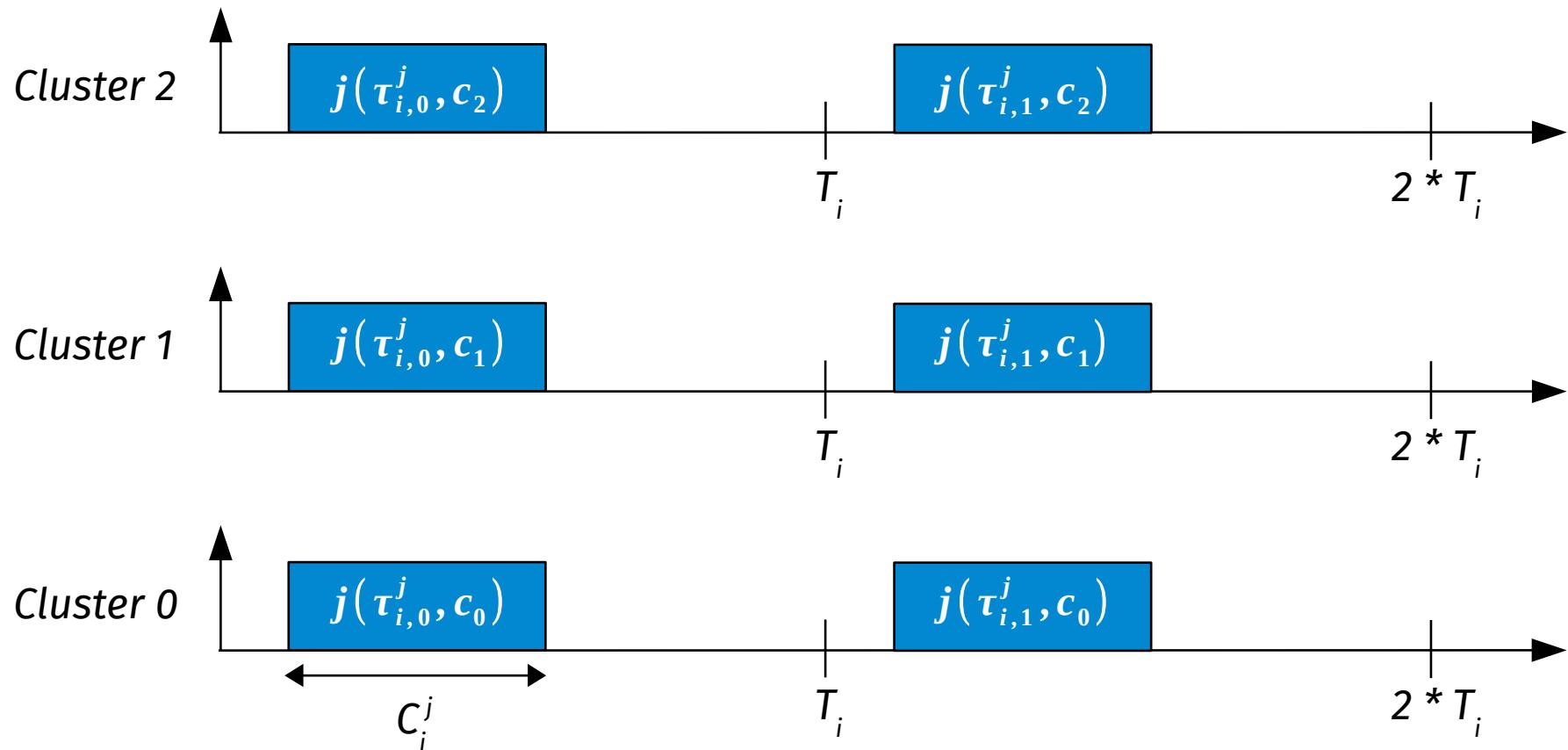
Niveau d'ordonnancement

- Un intervalle représente une *instance* de sous-tâche



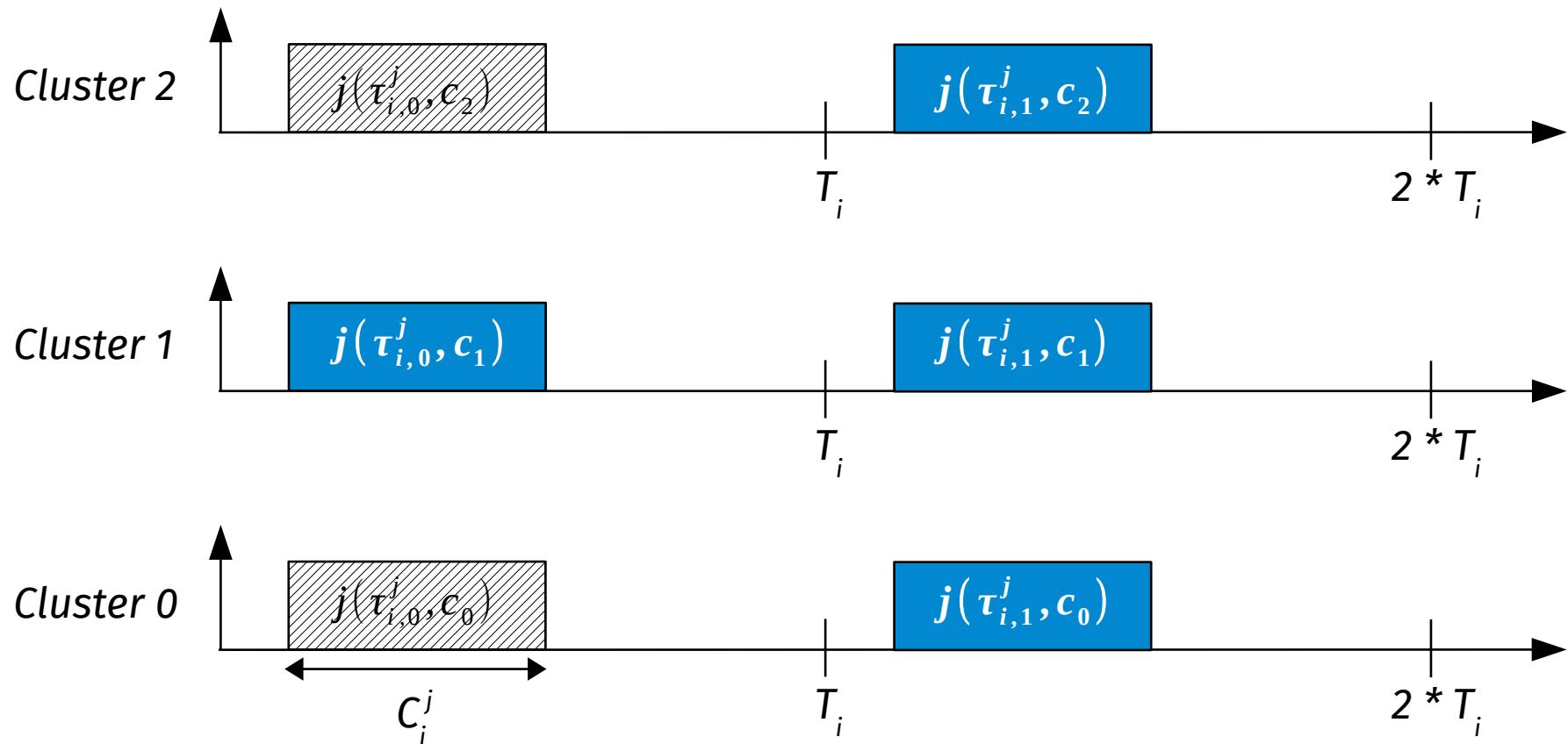
Niveau d'ordonnancement

- Un intervalle représente une **instance** de sous-tâche



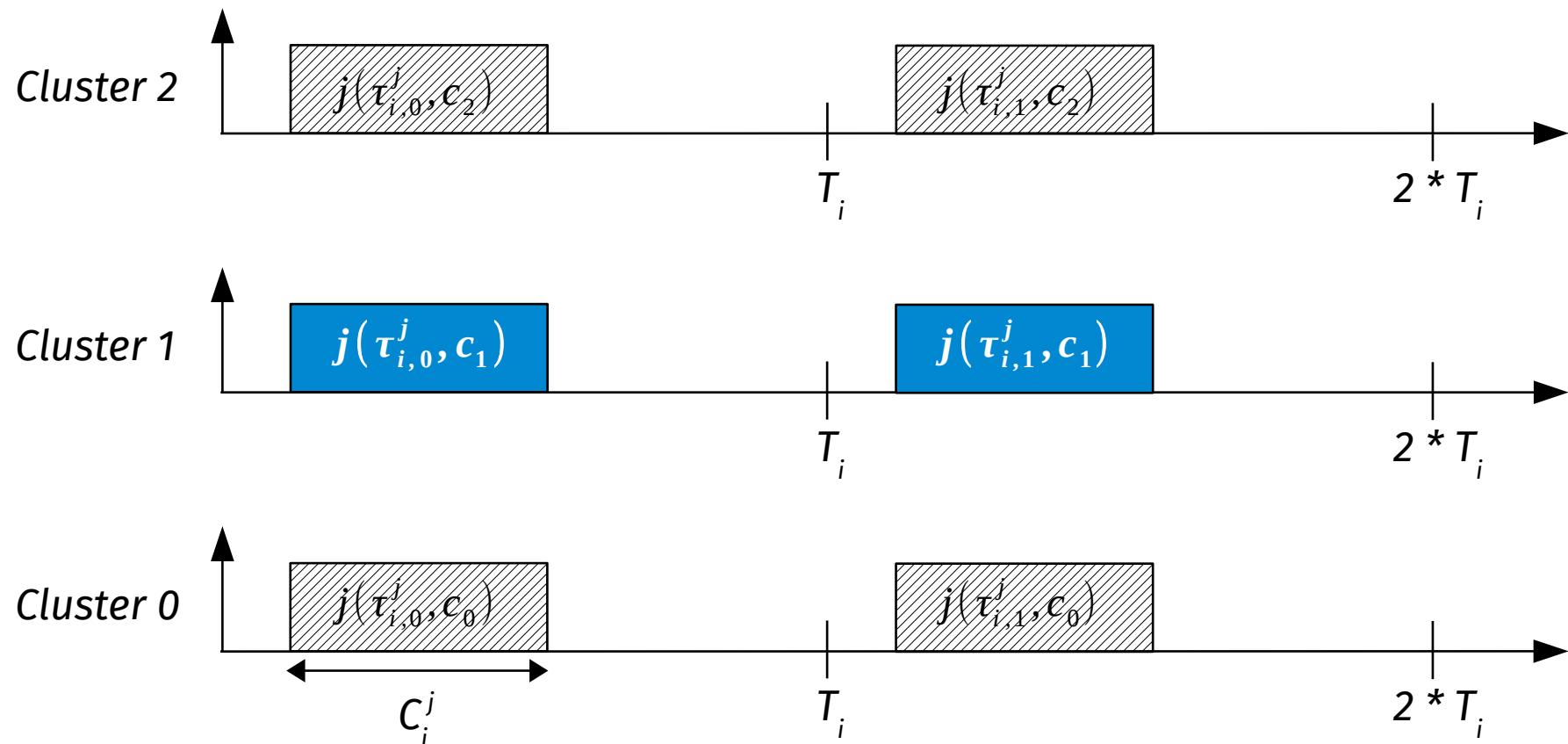
Niveau d'ordonnancement

- Un intervalle représente une **instance** de sous-tâche



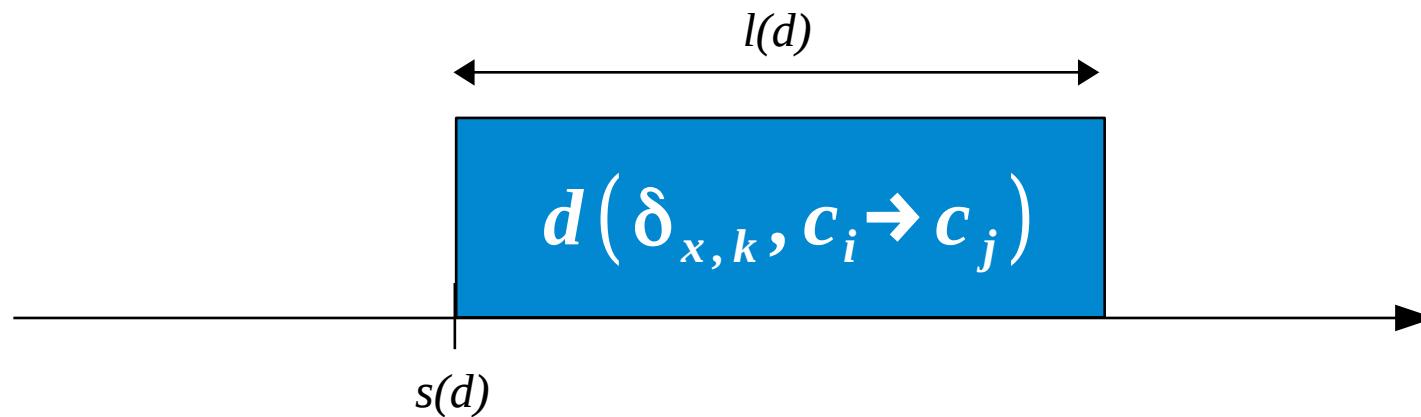
Niveau d'ordonnancement

- Un intervalle représente une **instance** de sous-tâche



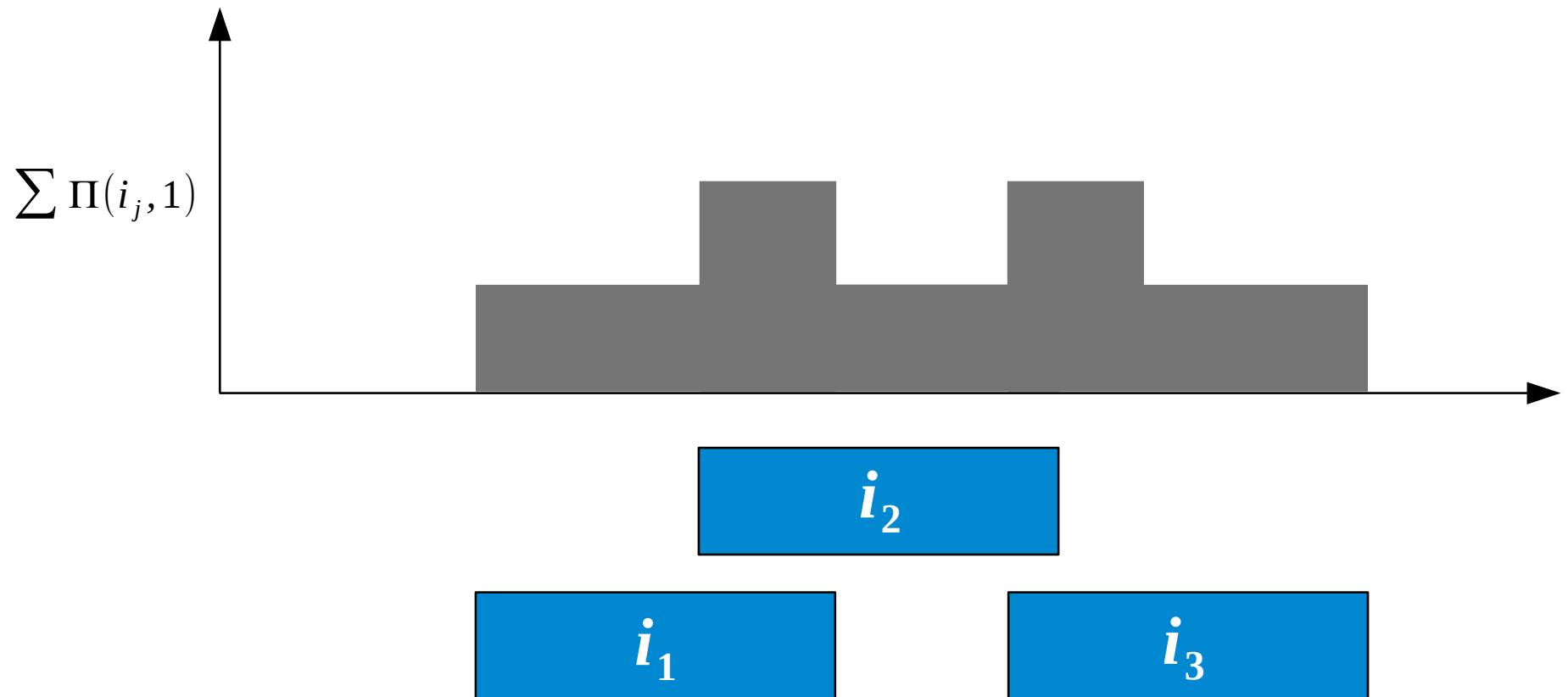
Modélisation des données

- 1 intervalle par **production** de donnée et par **canal** de communication

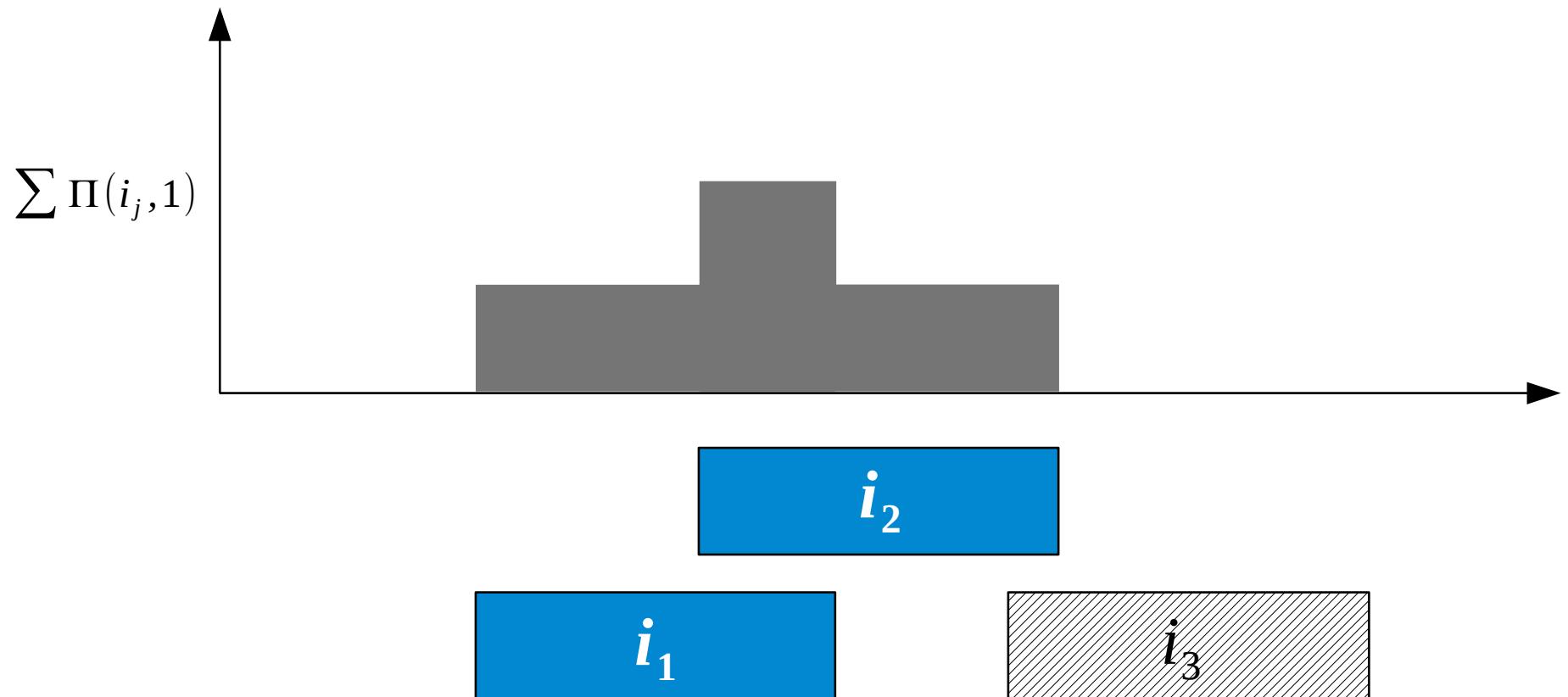


- $l(d)$ n'est **PAS** égal au temps nécessaire à l'envoi de $\delta_{x,k}$
- $l(d)$ est égal à la **durée du créneau** d'accès au NoC
- $s(d)$ encode l'**allocation de l'envoi** à un créneau

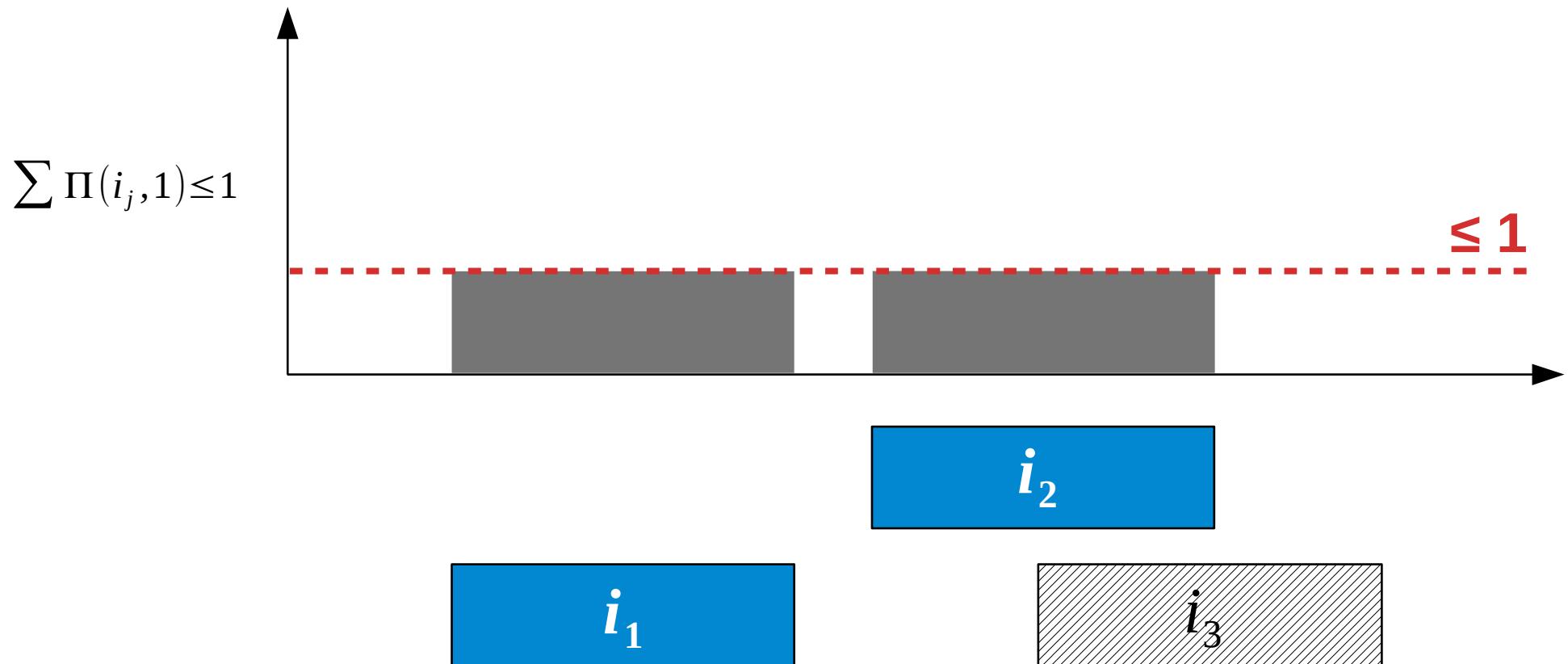
Fonction cumulative



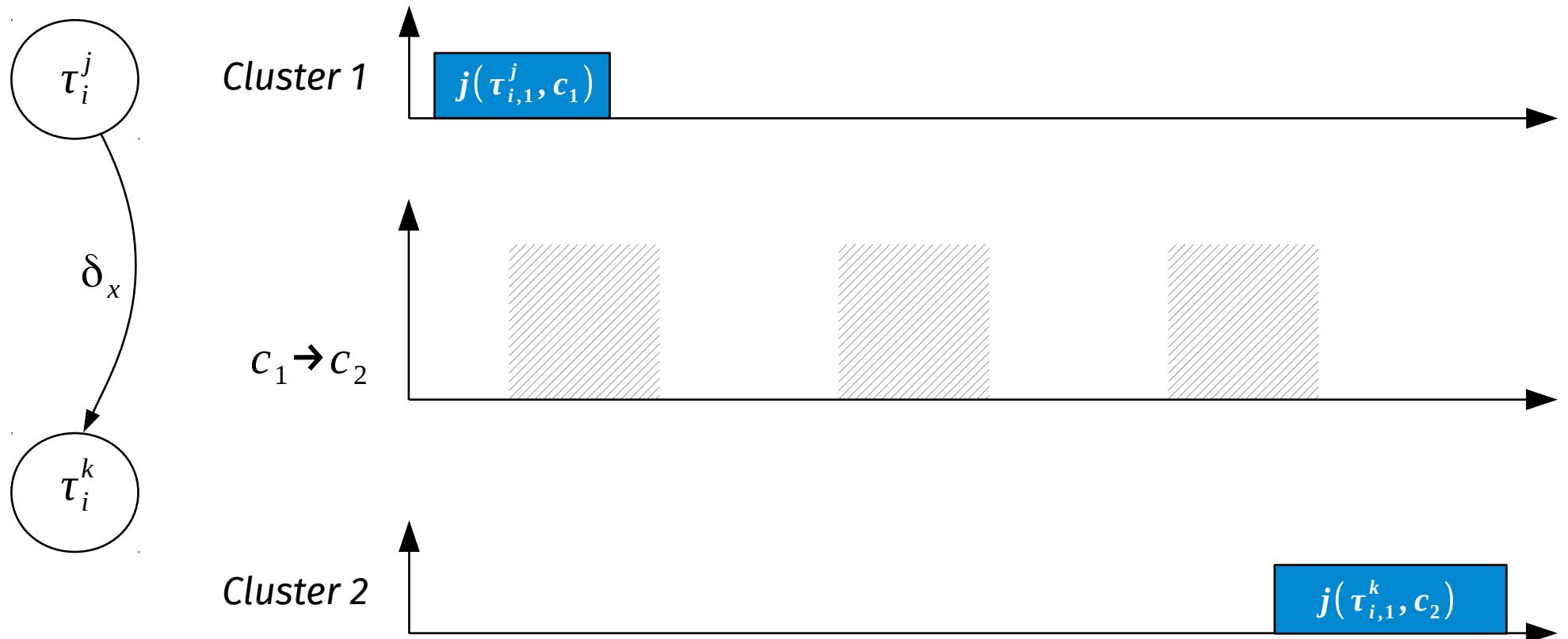
Fonction cumulative



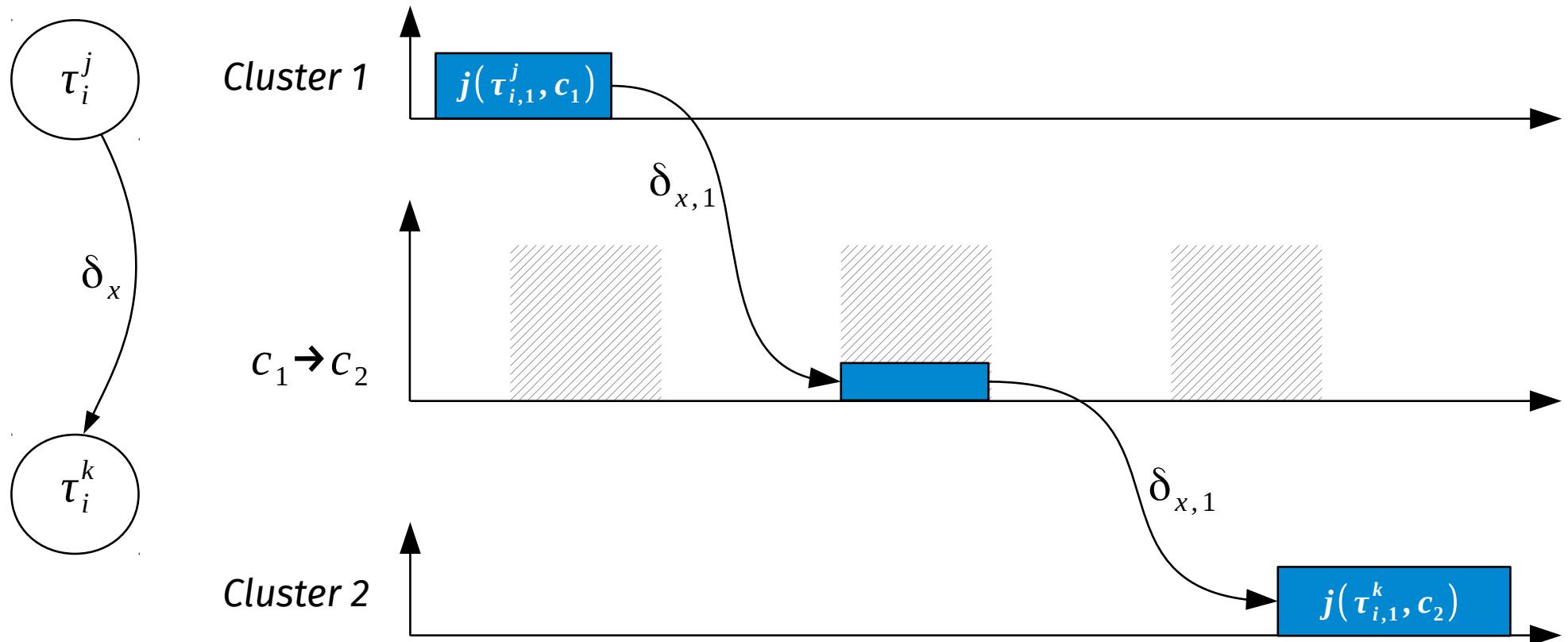
Fonction cumulative



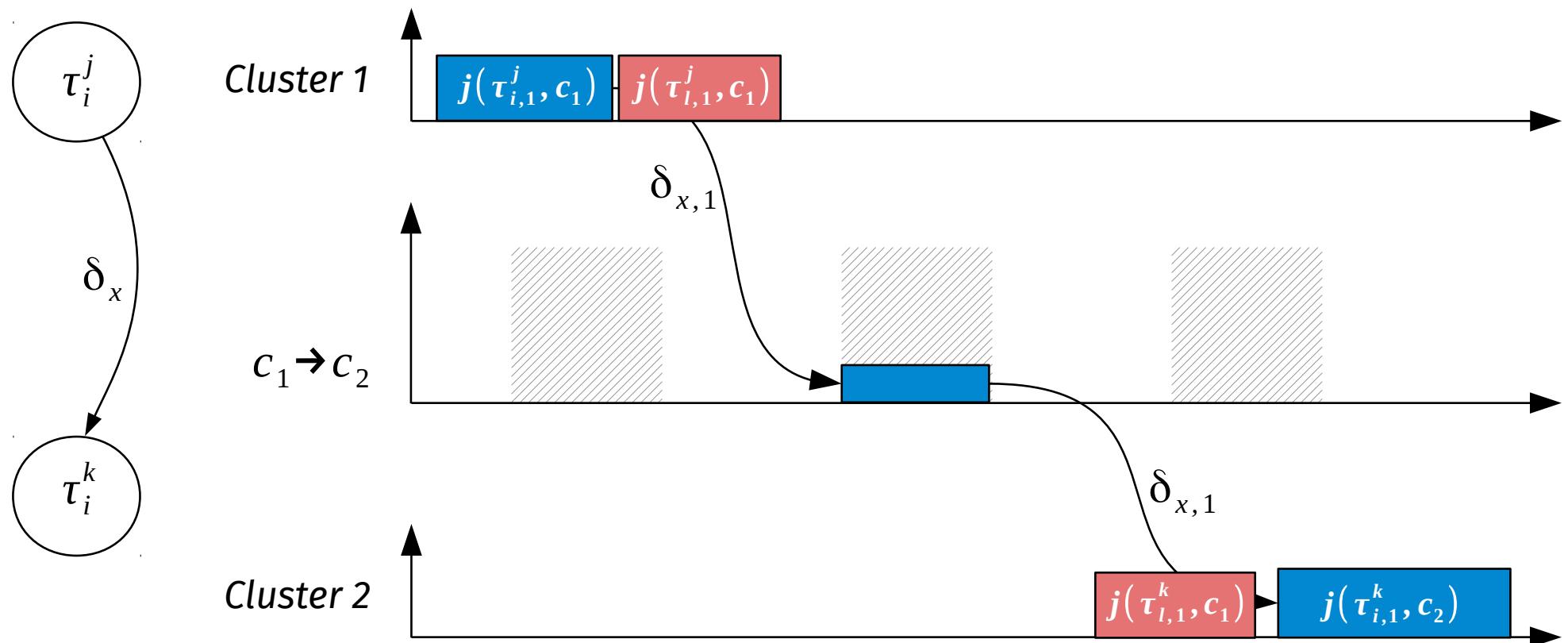
Modélisation des données



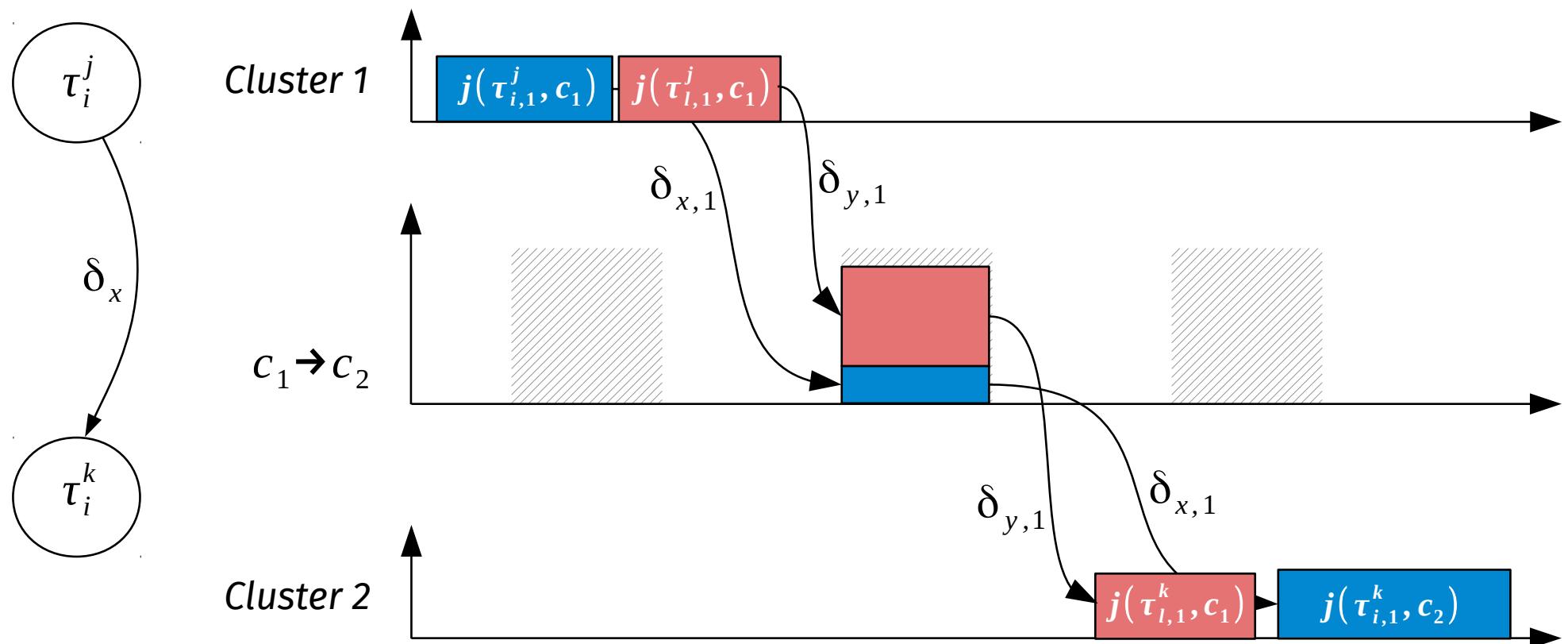
Modélisation des données



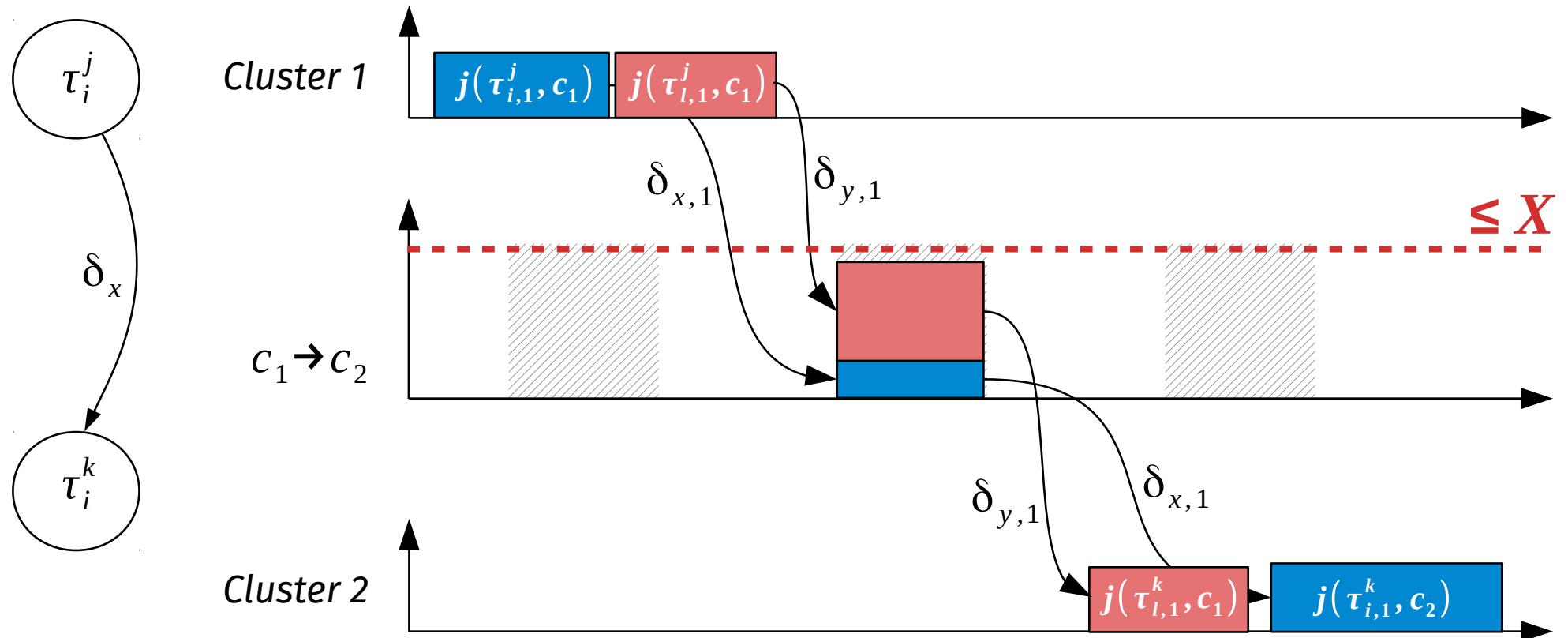
Modélisation des données



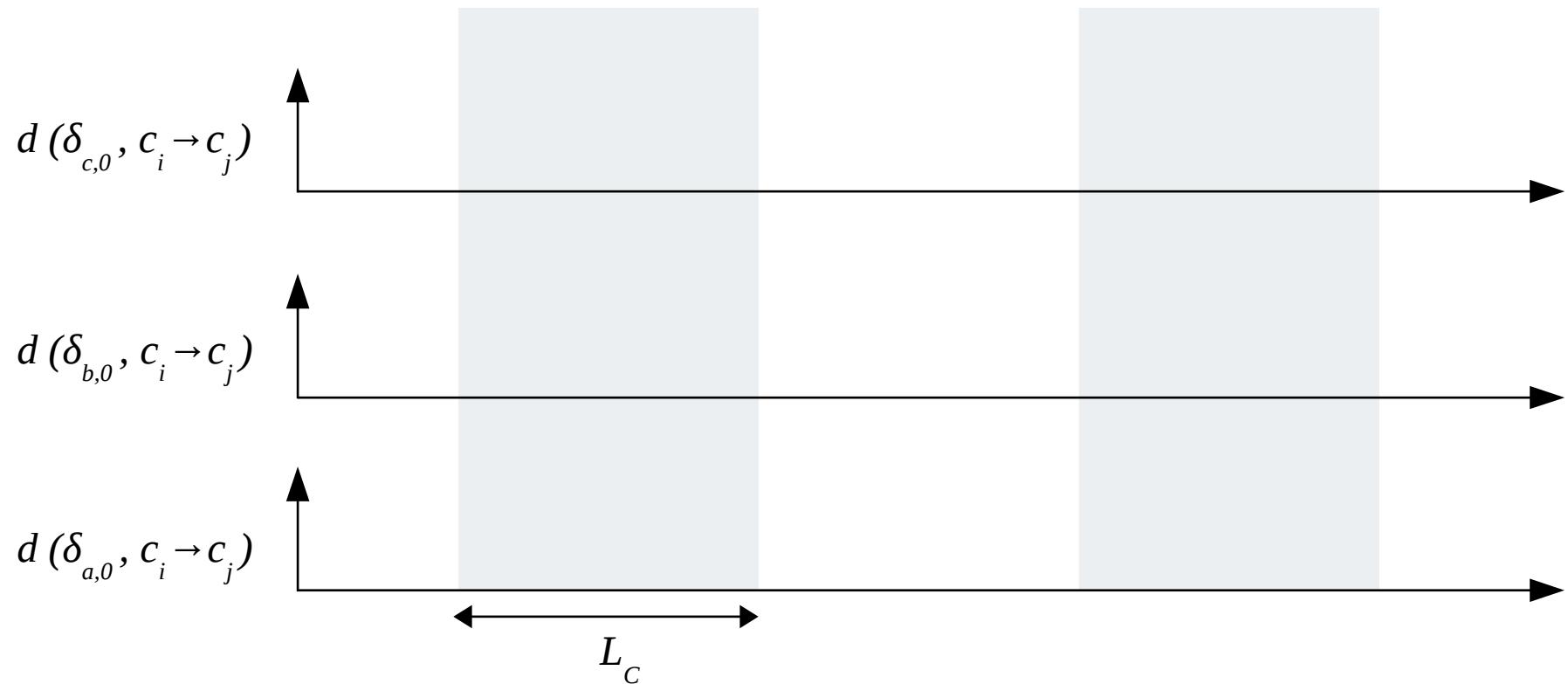
Modélisation des données



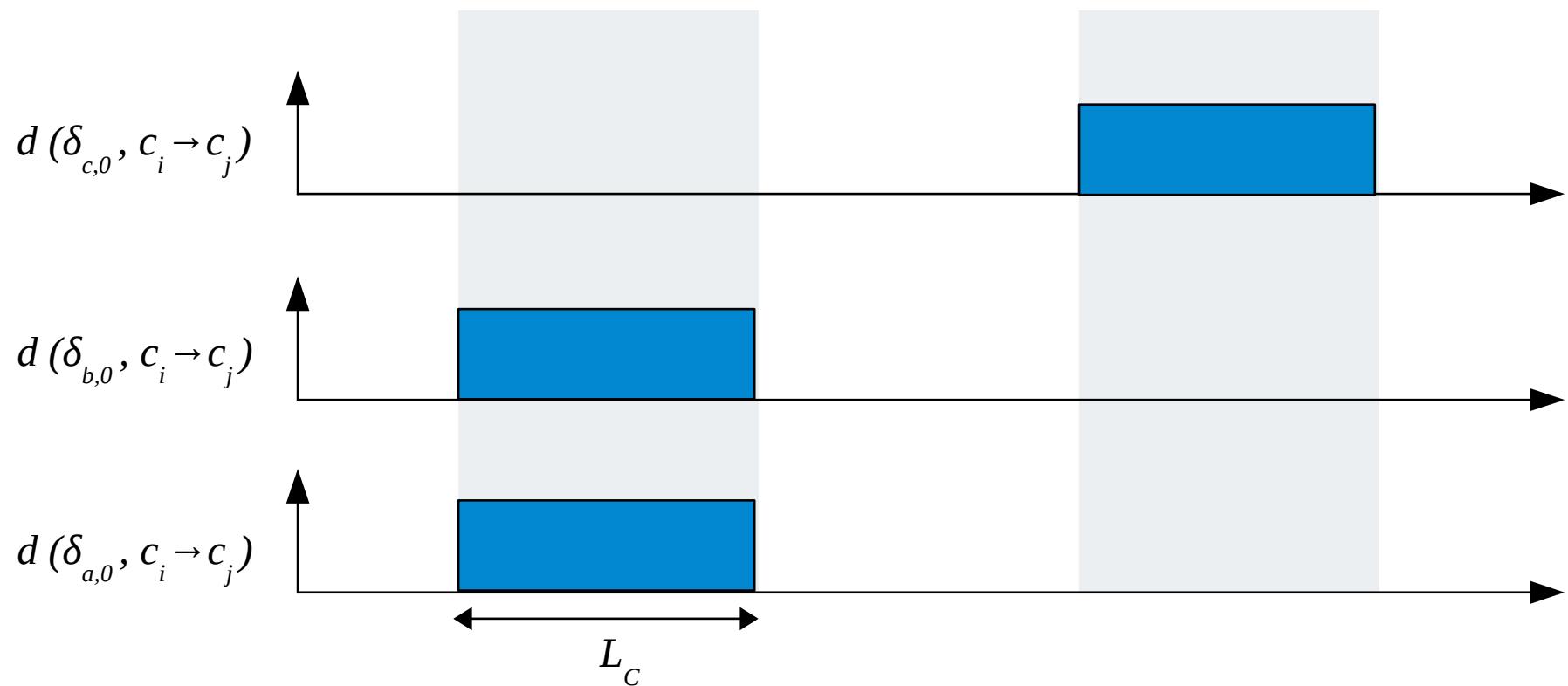
Modélisation des données



Modélisation des données



Modélisation des données



Modélisation des données

