Conception et mise en œuvre d'une plate-forme de pilotage de simulations numériques parallèles et distribuées

Nicolas Richart

LaBRI & INRIA Bordeaux – Sud-Ouest

20 janvier 2010

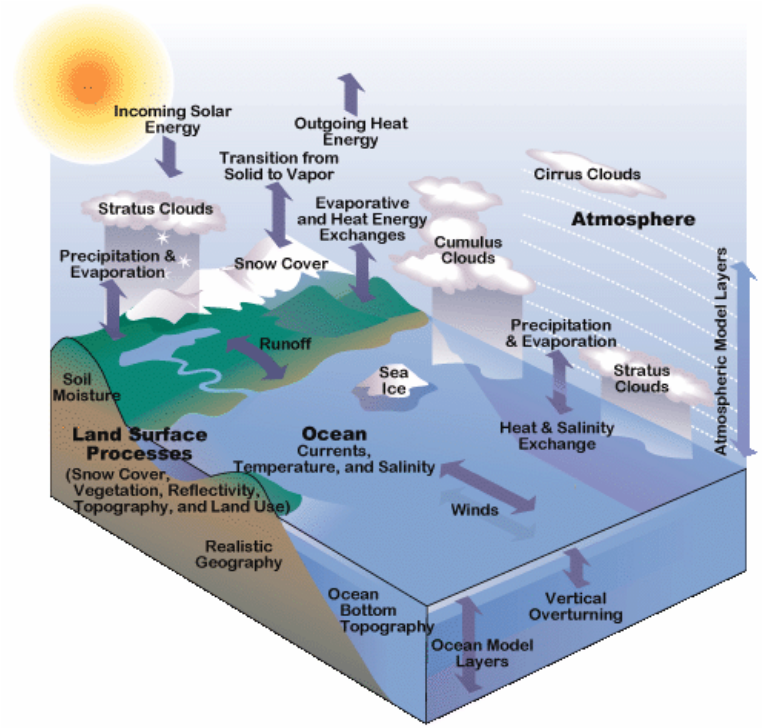
ANR
MASSIM

Plan

Plan

Simuler des phénomènes physiques complexes à l'aide d'ordinateurs

- Modélise des phénomènes complexes
  → mécanique des fluides, mécanique des solides, biologie moléculaire, ...
  → couplages de modèles physiques
- Couplage de codes
  → faire coopérer des codes existants
- Caractéristique des simulations
  → multi-physiques (ex. couplage fluide/structure)
  → multi-échelles (ex. couplage micro/macro en mécanique des solides)

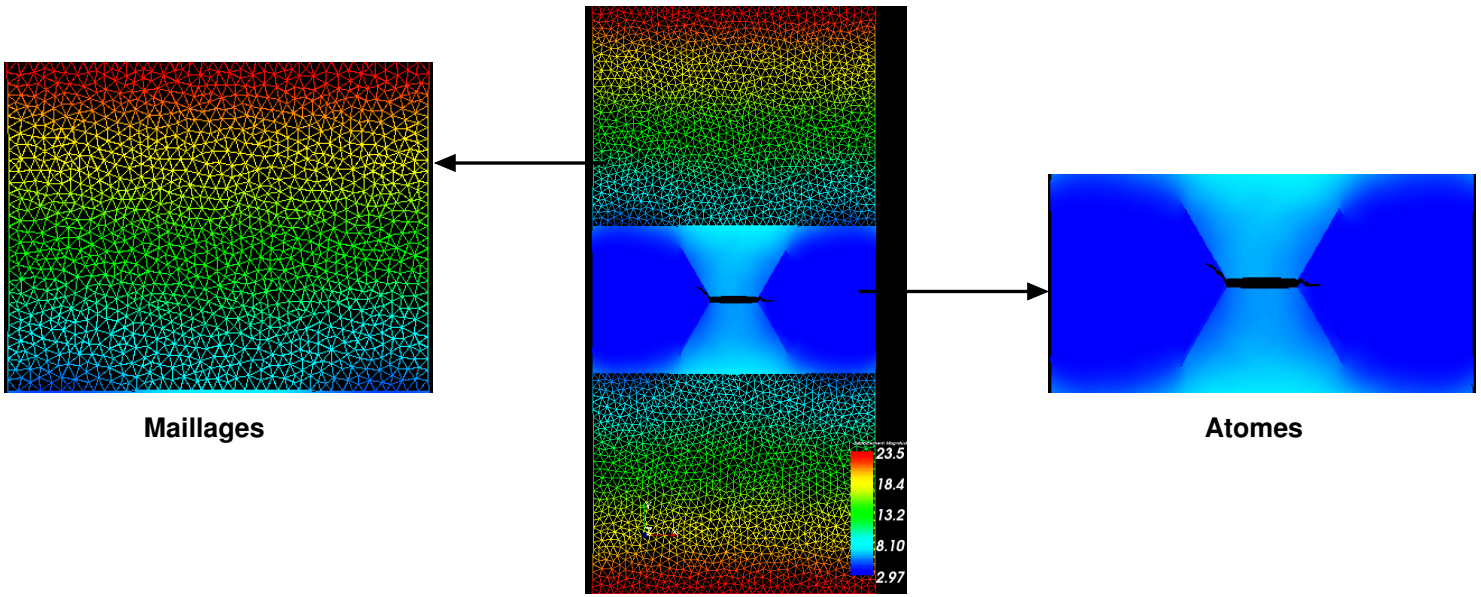


Couplage de modèles pour la climatologie.

Exemple de simulation couplée

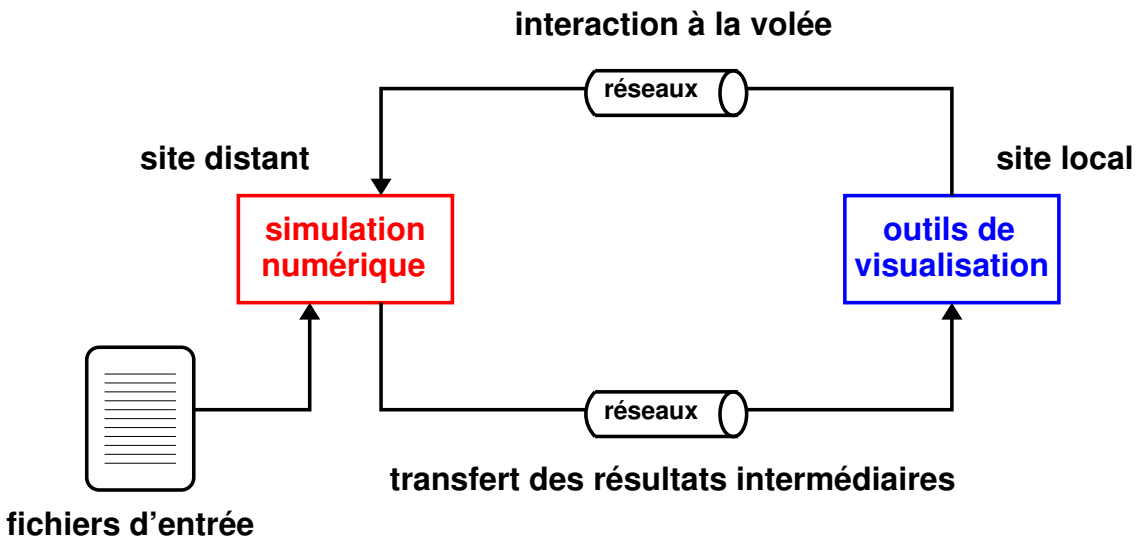
LibMultiScale : simulation couplée entre dynamique moléculaire et élasticité

- Plate-forme de couplage de codes multi-échelles
- Un code d'élasticité
  → code du Laboratoire de Simulation de la Mécanique des Solides (EPFL-ENAC-IIS-LSMS)
  → maillage avec une donnée représentant le déplacement
- Un code de dynamique moléculaire
  → LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) développé au Sandia National Labs
  → atomes (points) avec une position initiale et une position courante
- Visualiser le déplacement

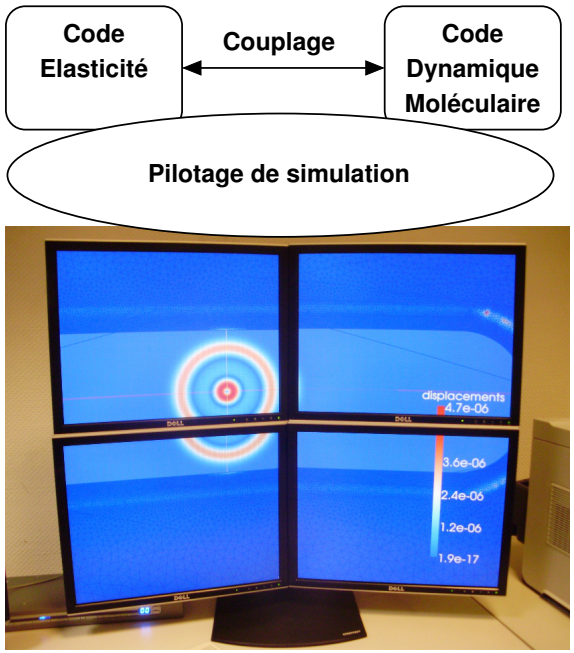


Analyse des résultats d'une simulation

- Exécution en mode "batch" et post-traitement
  → la simulation génère des fichiers de résultats
  → les fichiers sont analysés en fin de simulation
  → suivant le résultat la simulation est ré-exécutée avec un nouveau jeu de paramètres
- Le pilotage de simulation
  → les données sont visualisées en cours de simulation
  → si le résultat n'est pas bon, les paramètres de la simulation sont modifiés "à la volée"



Le pilotage de simulation



Intérêts

- Comprendre la dynamique
  → surveiller l'évolution d'une simulation
- Analyse de sensibilité
  → avoir un retour direct sur les simulations
  → pouvoir modifier les paramètres des simulations

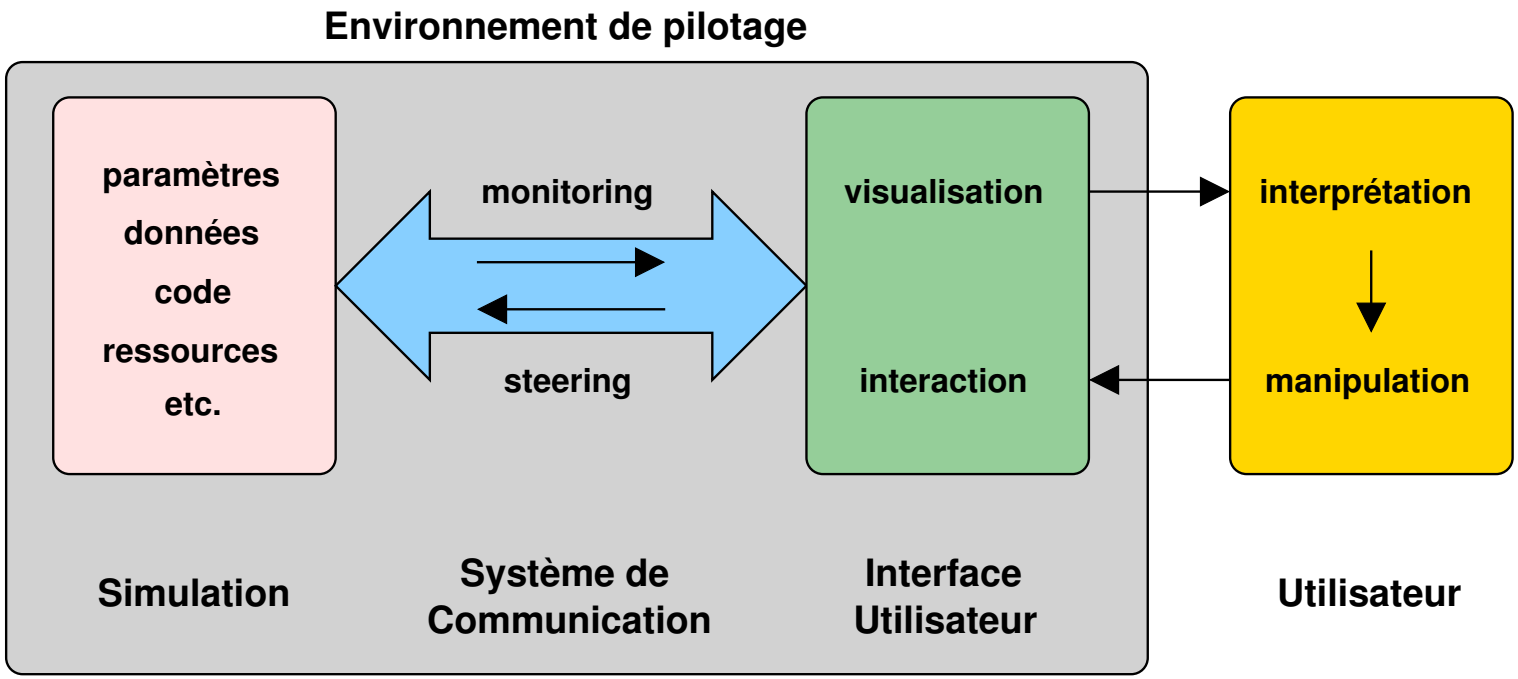
Difficultés

- Modéliser les simulations
  → flot d'exécution, données communes aux codes, mais de natures différentes
- Coordonner les opérations de pilotage
  → garantir la cohérence des opérations de pilotage, que ce soit entre des codes couplés, ou dans chacun des codes
- Performance
  → ne pas trop perturber les simulations

Plan

Les trois composantes logicielles

- Simulation numérique
  → boucle de calcul en temps
- Système de communication
  → réalisation des transferts de monitoring et de steering
- Interface utilisateur
  → visualisation + interaction



Environnement de pilotage

La modélisation des simulations

- Boucle de calcul (ex. CUMULVS)
  → un ou plusieurs points d'instrumentation placés dans la boucle principale
- Modules/Composants (ex. SCIRun)
  → modélisation des fonctionnalités des codes en modules
  → représentation faite dans les PSE (Problem Solving Environement)
- Arbres de tâches hiérarchiques (ex. EPSN)
  → représentation des codes en un ensemble de tâches imbriquées

Stratégie pour assurer la cohérence des traitements de pilotage

- Modèle data-flow
- Synchronisation forte
- Synchronisation faible ou planification

Les principaux environnements existants

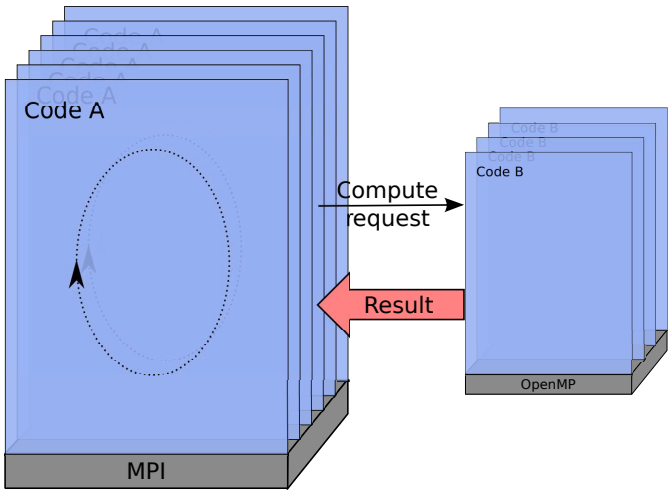
Environnement	Simulation	Modélisation	Pilotage
SCIRun (PSE)	mémoire partagée	module	data-flow
Cactus (PSE)	SPMD/distribuée	module	data-flow
RealityGRID	SPMD/couplée	boucle de calcul	synchro. forte
VISIT	SPMD	point	synchro. forte
CUMULVS	SPMD	boucle de calcul	synchro. faible
EPSN	SPMD	arbre en tâches	synchro. faible

- EPSN est une bonne solution
  → thèse de A. Esnard (2005)
  → pas de support des simulations couplées

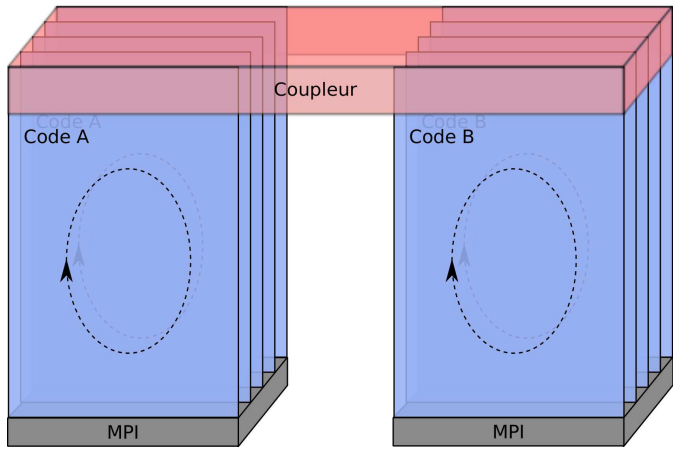
Plan

Le pilotage de simulations numériques parallèles et distribuées

- Des “legacy codes”
- Les simulations couplées
  → simulation Client/Serveur
  → simulations M-SPMD (Multiple-SPMD)



Code Client/Serveur



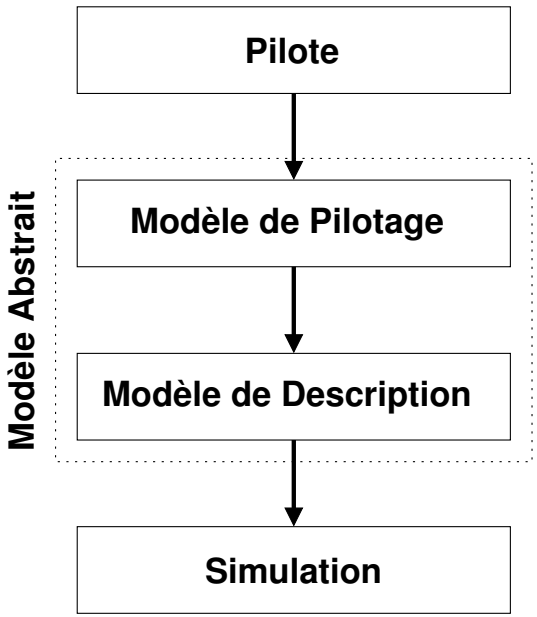
Code M-SPMD

Contributions

- Modélisation des simulations
  → modèle de représentation unique pour les simulations visées
  → modèle hiérarchique en tâches (MHT)
  → modèle pour les données distribuées
- Cohérence des traitements
  → définir la cohérence d'un traitement
  → coordonner les traitements dans la simulation
  → assurer cette cohérence tout au long des traitements
- Performance

Recherche d'une approche générique pour le pilotage de simulations

- Modèle de description
  → description en arbre de tâches de la structure d'un programme (MHT)
  → modèle basé sur l'instrumentation du code source
  → description simple des données, support + variables associées
- Modèle de pilotage
  → pilotage par des requêtes
  → association des interactions possibles aux tâches du MHT

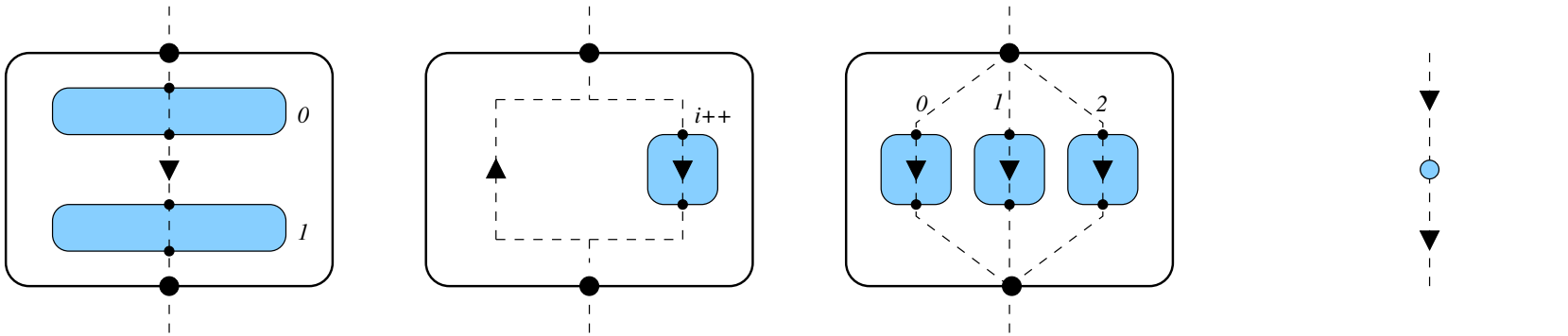


Le modèle doit indiquer à l'environnement de pilotage où, quand et comment interagir de manière cohérente avec le code de simulation.

Plan

Description des programmes en arbre de tâches

- Modèle simple basé sur des tâches hiérarchiques
  → tâche contenant des sous-tâches
- Quatre types de tâches de base (simple, boucle, conditionnelle, point)
  → capturer le flot d'exécution

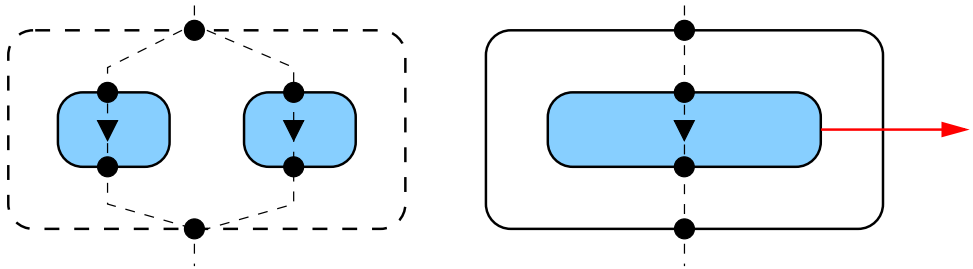


(a) tâche composée **(b) tâche en boucle** **(c) tâche conditionnelle** **(d) tâche en point**

- Description à grain moyen du code de simulation
  → annoter dans le source uniquement les tâches pertinentes
- Une tâche hiérarchique englobant toute la simulation

Description des programmes distribués

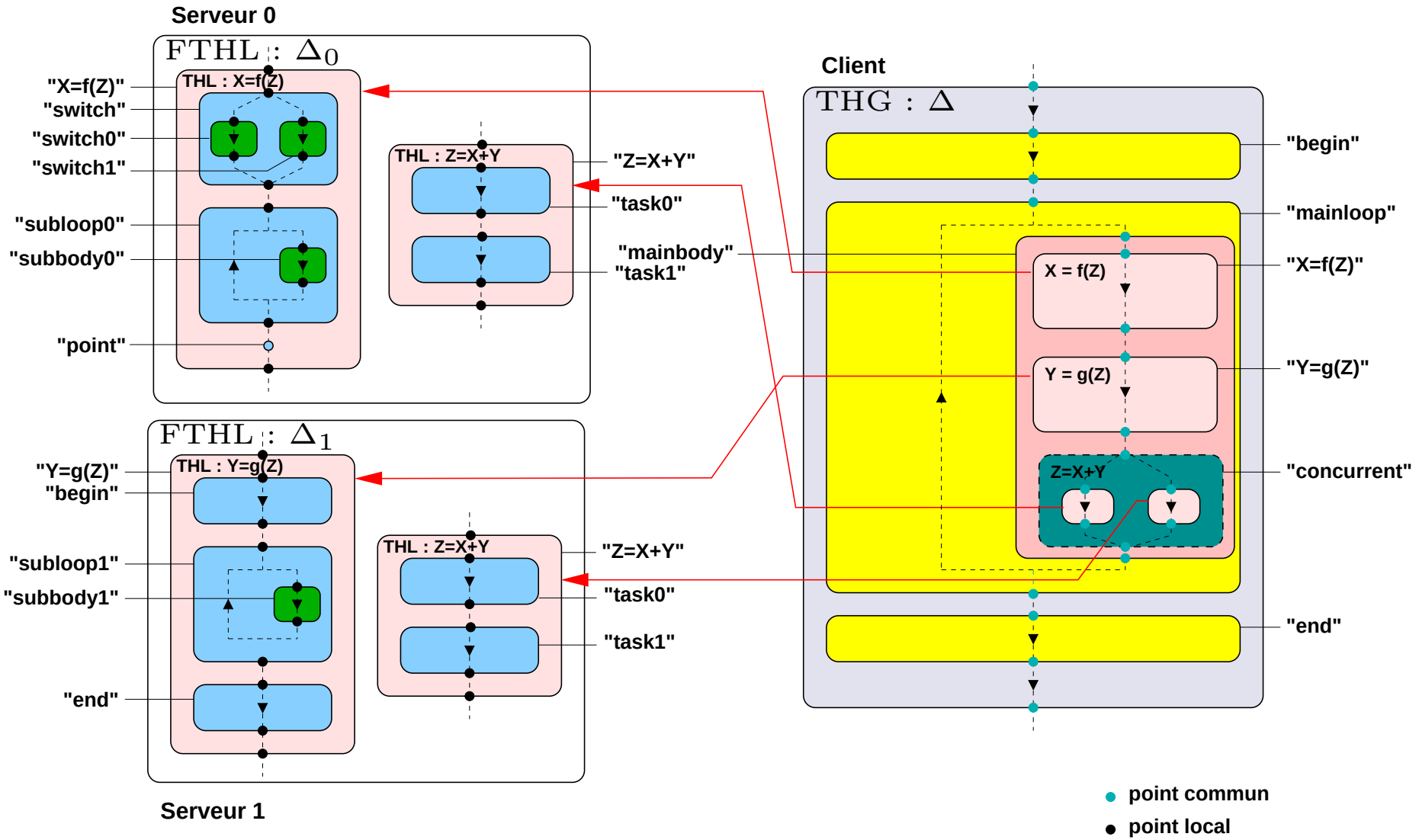
- Deux types de tâches pour les simulations distribuées



(a) tâche concurrente (b) tâche distante

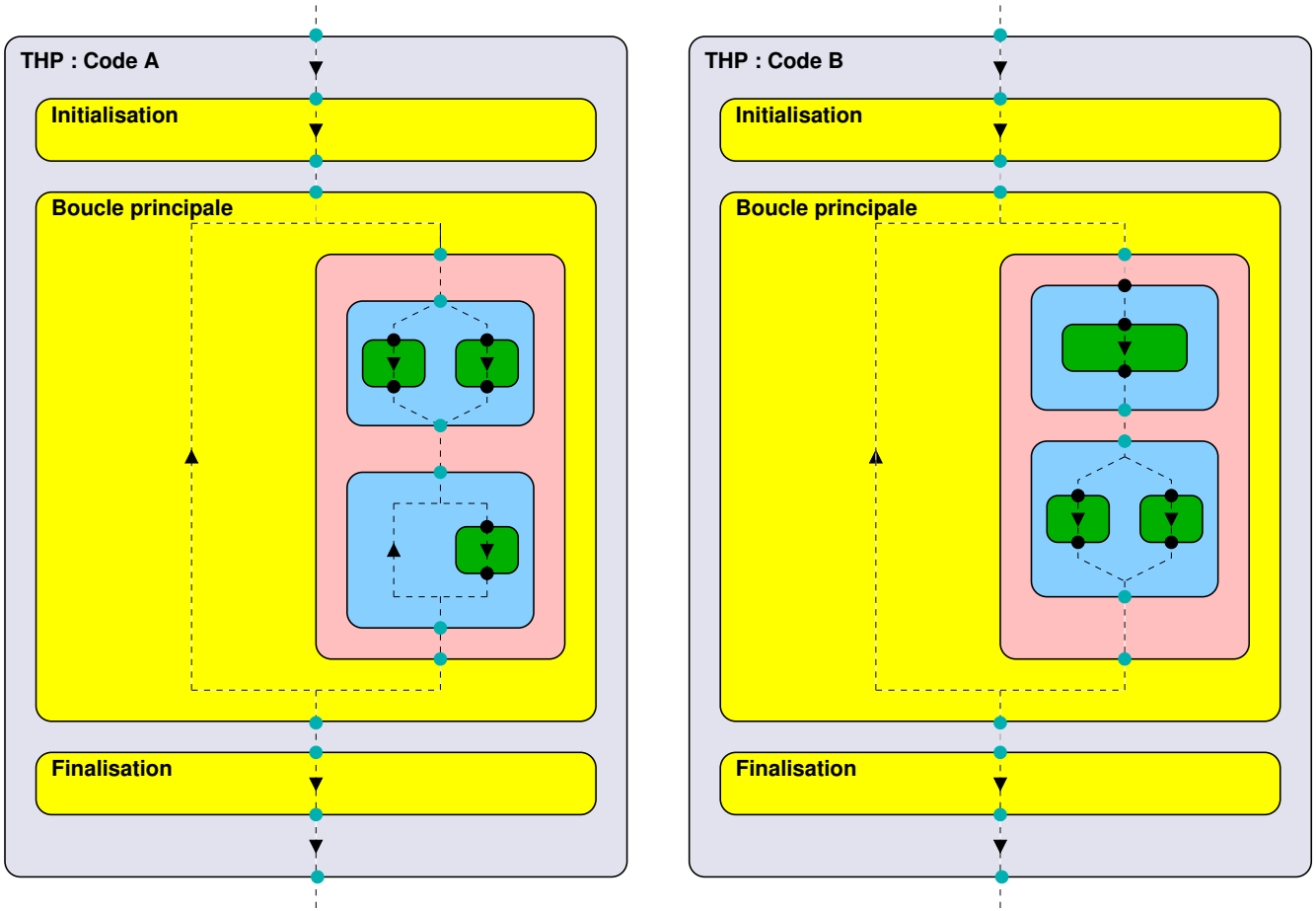
- Une tâche hiérarchique englobant la boucle principale de la simulation (THG)
  → le code client dans le cas de simulations Clients/Serveurs
  → les parties communes dans le cas des simulations M-SPMD
- Une forêt de tâches hiérarchiques pour les méthodes distantes

Exemple d'un code Client/Serveur



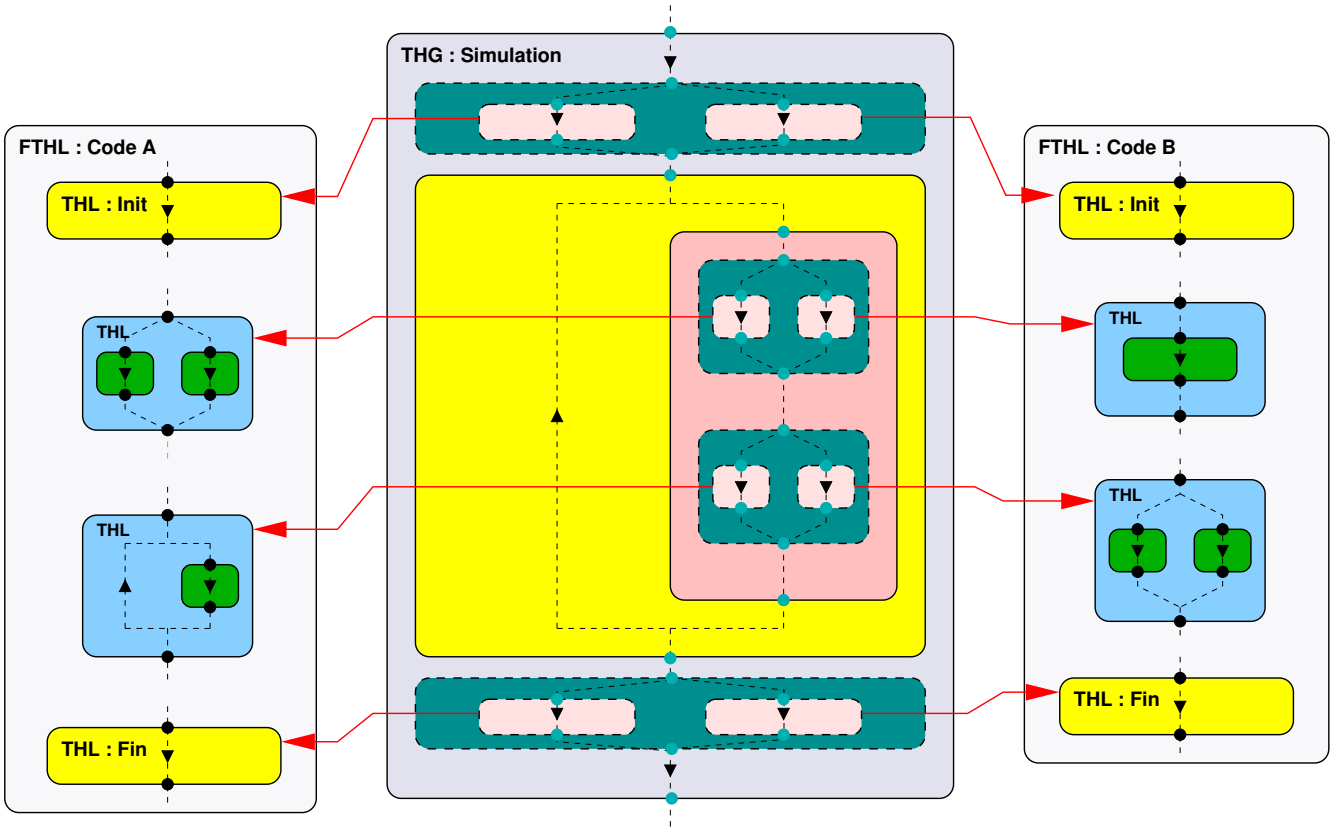
Exemple de représentation d'une simulation distribuée

Cas d'une simulation M-SPMD



Exemple de représentation d'une simulation distribuée

Cas d'une simulation M-SPMD



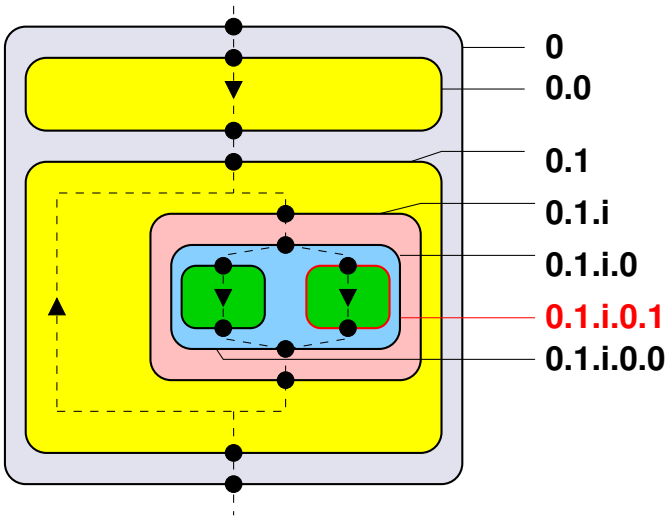
Un modèle unifié pour les simulations Clients/Serveurs et M-SPMD

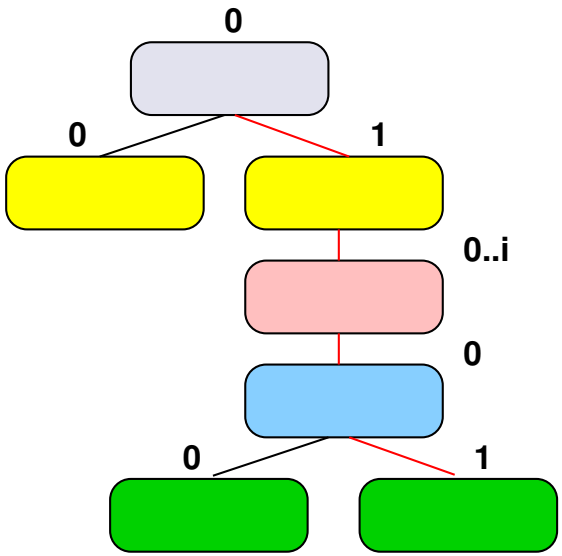
Système de datation associé aux tâches hiérarchiques

Se repérer précisément dans le flot d'exécution et planifier des traitements

Pour une simulation parallèle

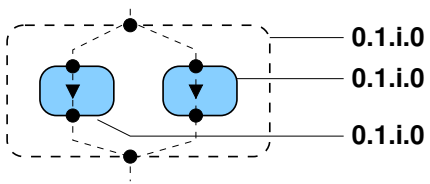
- Date de tâche
  → concaténation des indices des tâches imbriquées
- Date de point
  → date de tâche + position en début ou en fin de tâche
  → associée aux points d'instrumentation
  → relation d'ordre stricte et totale pour comparer les dates



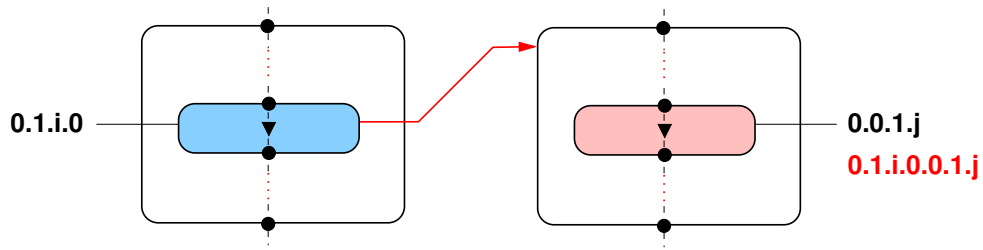


Pour une simulation distribuée

- Date restreinte
  → correspond aux dates dans un code
- Date complète
  → correspond aux dates dans une simulation
- Masque de date
  → pseudo date permettant de définir un ensemble de dates
  ex. $0.1.\widetilde{2}.0 = \{0.1.2.0,\ 0.1.4.0,\ \cdots,\ 0.1.2k.0\}$



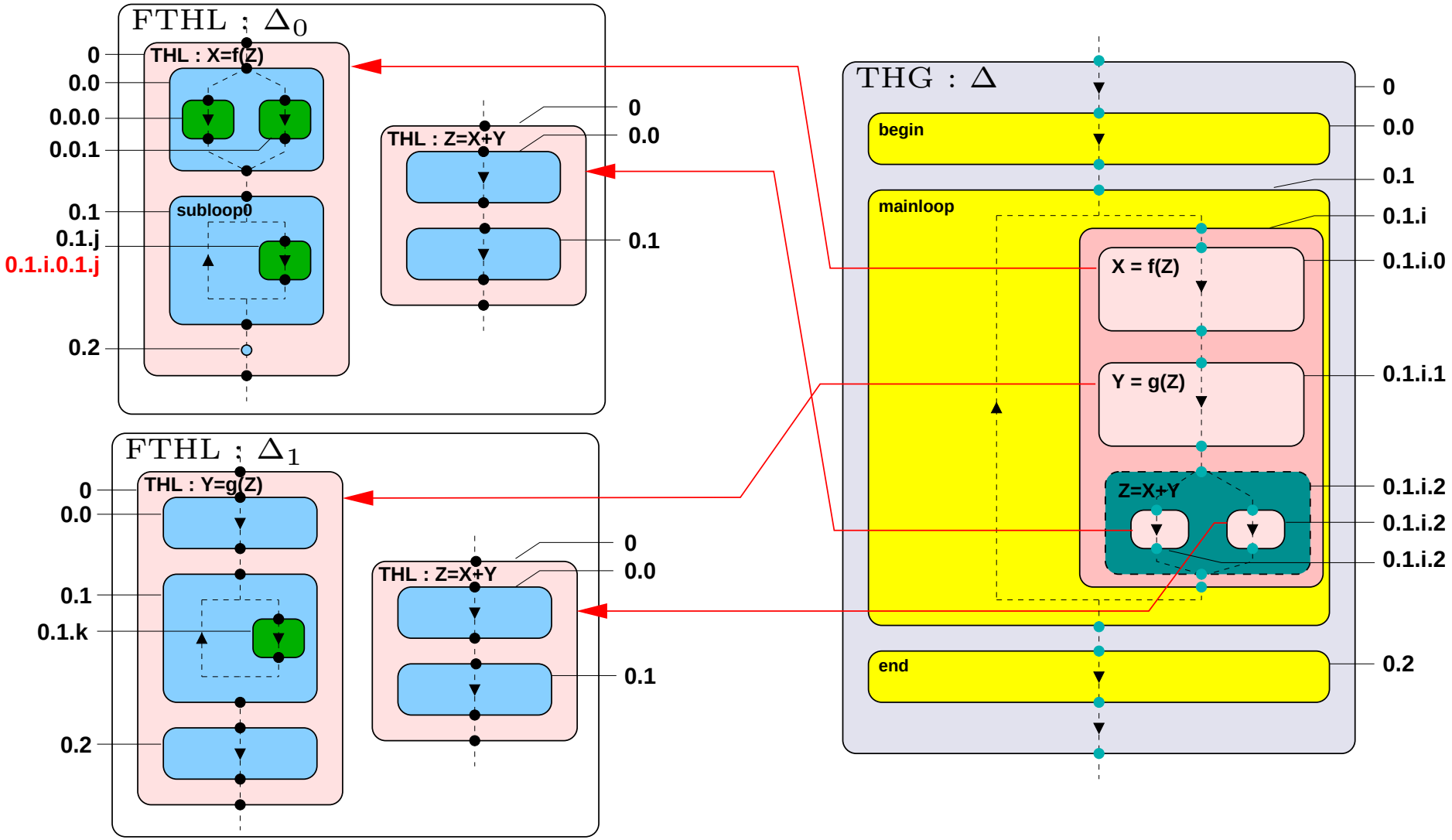
Tâche concurrente



Tâche distante

Exemple de date pour une simulation distribuée

Dates associées à la simulation Client/Serveur précédente



Description des données

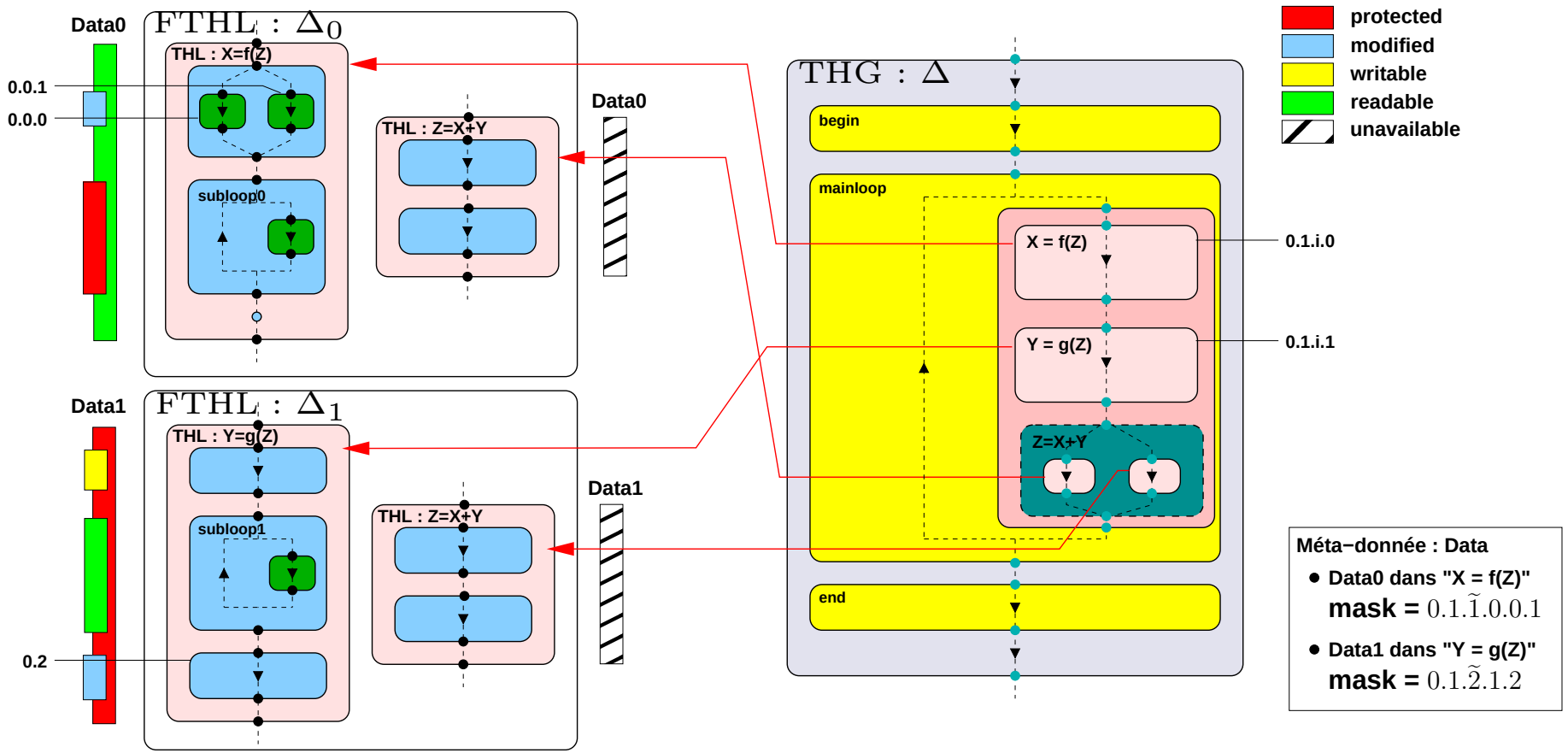
Les données simples (dans les codes)

- Support
  → grilles, maillages, points, paramètres
- Ensemble de variables associées
- Contexte d'accès aux variables
  → associé aux tâches
  → *readable, writable, modified, protected, unavailable*
- Révision
  → la date de la tâche à laquelle la donnée a été générée

Les méta-données (données transversales aux codes)

- Liste de variables
  → sous-ensemble de variables des données des codes
  → nom des codes d'origine des variables
- Informations d'accessibilité
  → même contexte que ceux des variables des données simples pour *readable, writable, protected, unavailable*
  → un masque de date définissant l'ensemble des révisions cohérentes entre elles

Données associées à la simulation Client/Serveur précédente



Plan

Modèle de pilotage par les requêtes

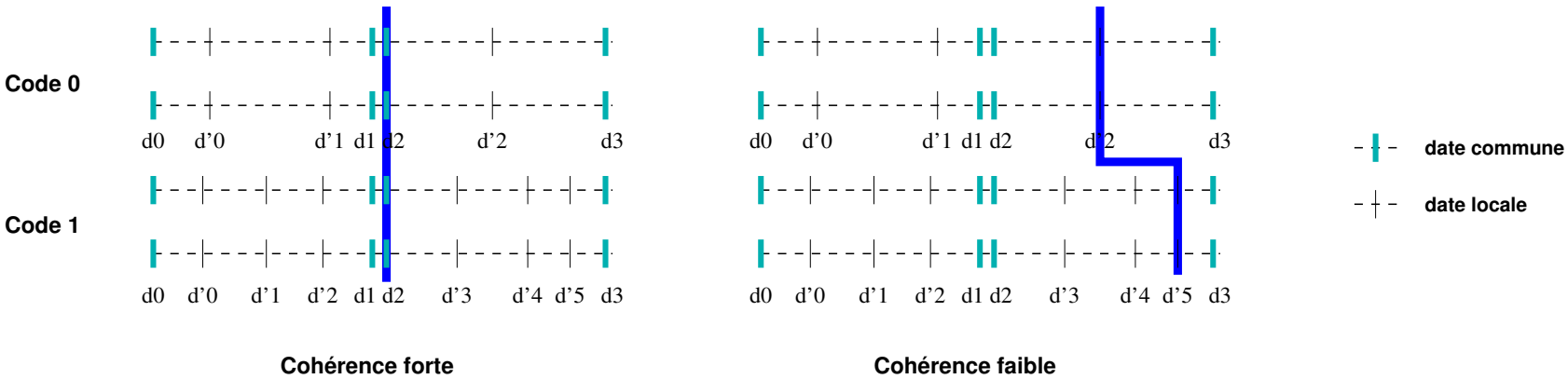
- Requêtes simples
  → get, put, action, play/step/pause
- Requêtes répétées
  → envoi périodique (getp), actions répétées

Le cycle de vie des requêtes

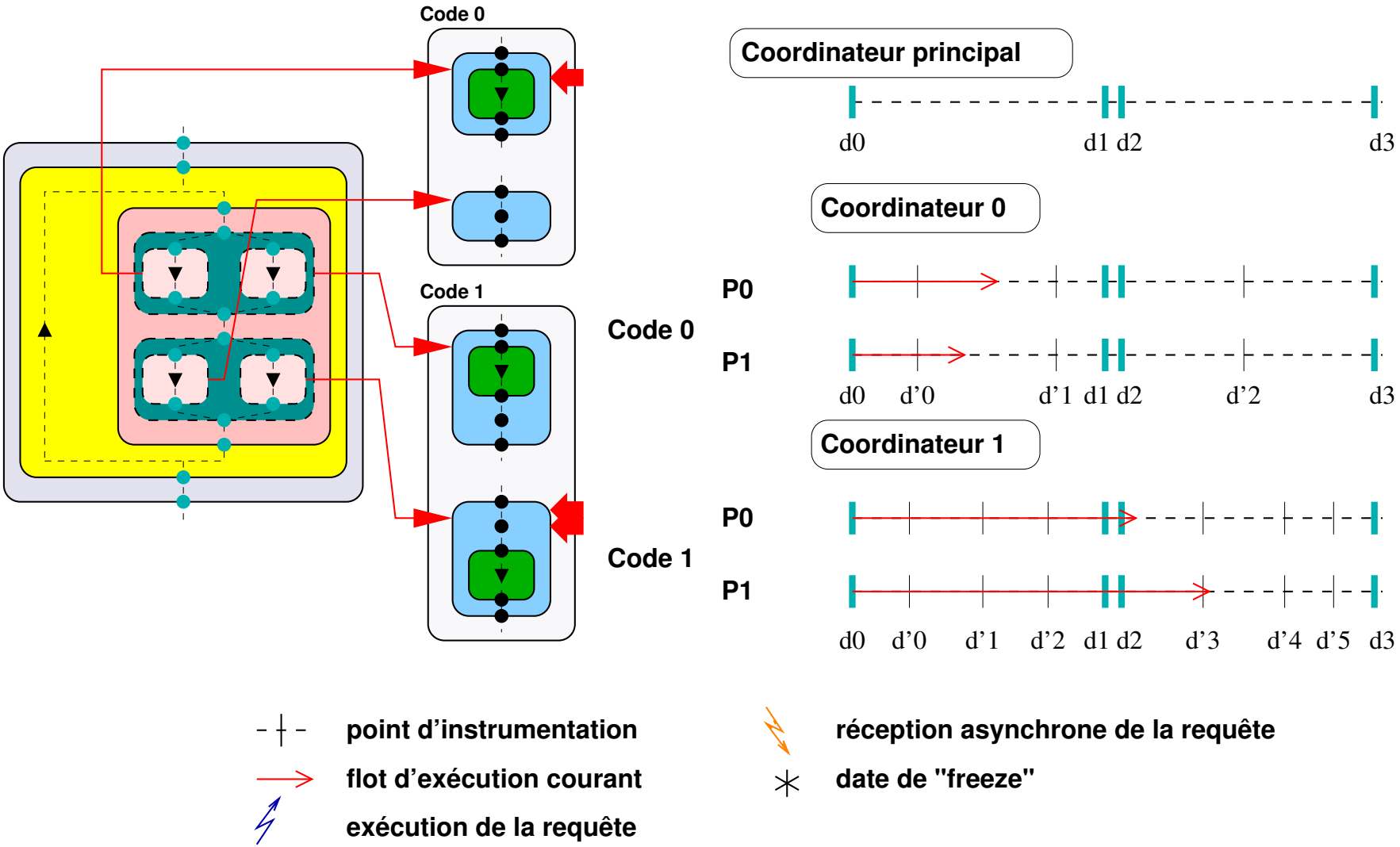
- Réception d'une requête
- Coordination
  → coordination globale à la simulation
  → coordination locale aux codes
- Vérification des conditions locales
  → conditions propres aux différentes types de requêtes
- Exécution de la requête
- Acquittement

La cohérence des traitements côté simulation

- Pour une simulation parallèle
  → traitement exécuté à la même date pour tous les processus
  → condition locale remplie (ex. pour des données : accessibilité)
- Pour une simulation distribuée
  → pas de date commune à tous les processus
  → **cohérence forte** : traitement exécuté sur une date commune à tous les codes
  → **cohérence faible** : traitement exécuté à un même pas de temps global mais pas forcément à la même date
  → choix de la cohérence d'après les besoins des traitements de pilotage

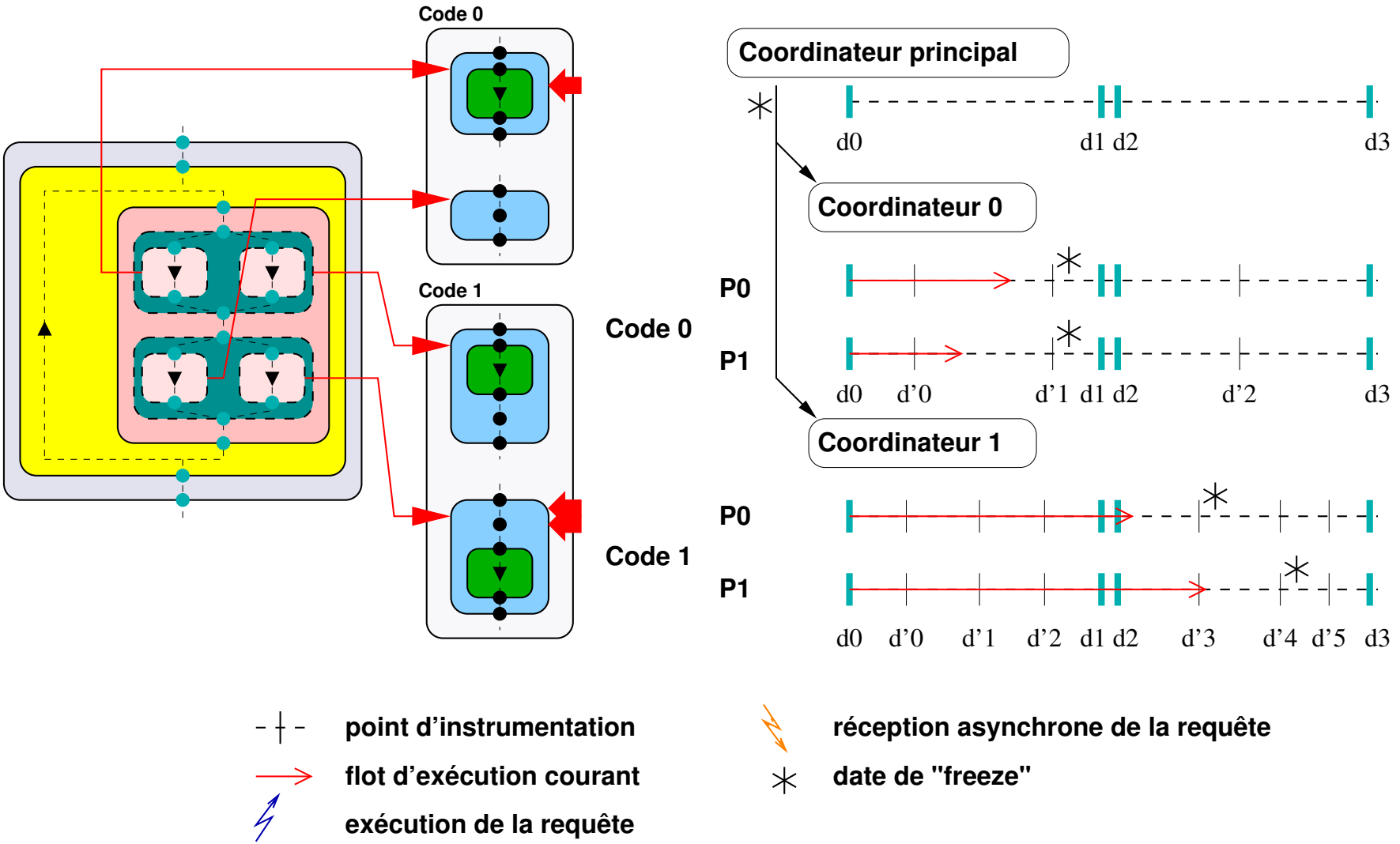


Planifier une date de traitement afin de garantir la cohérence temporelle



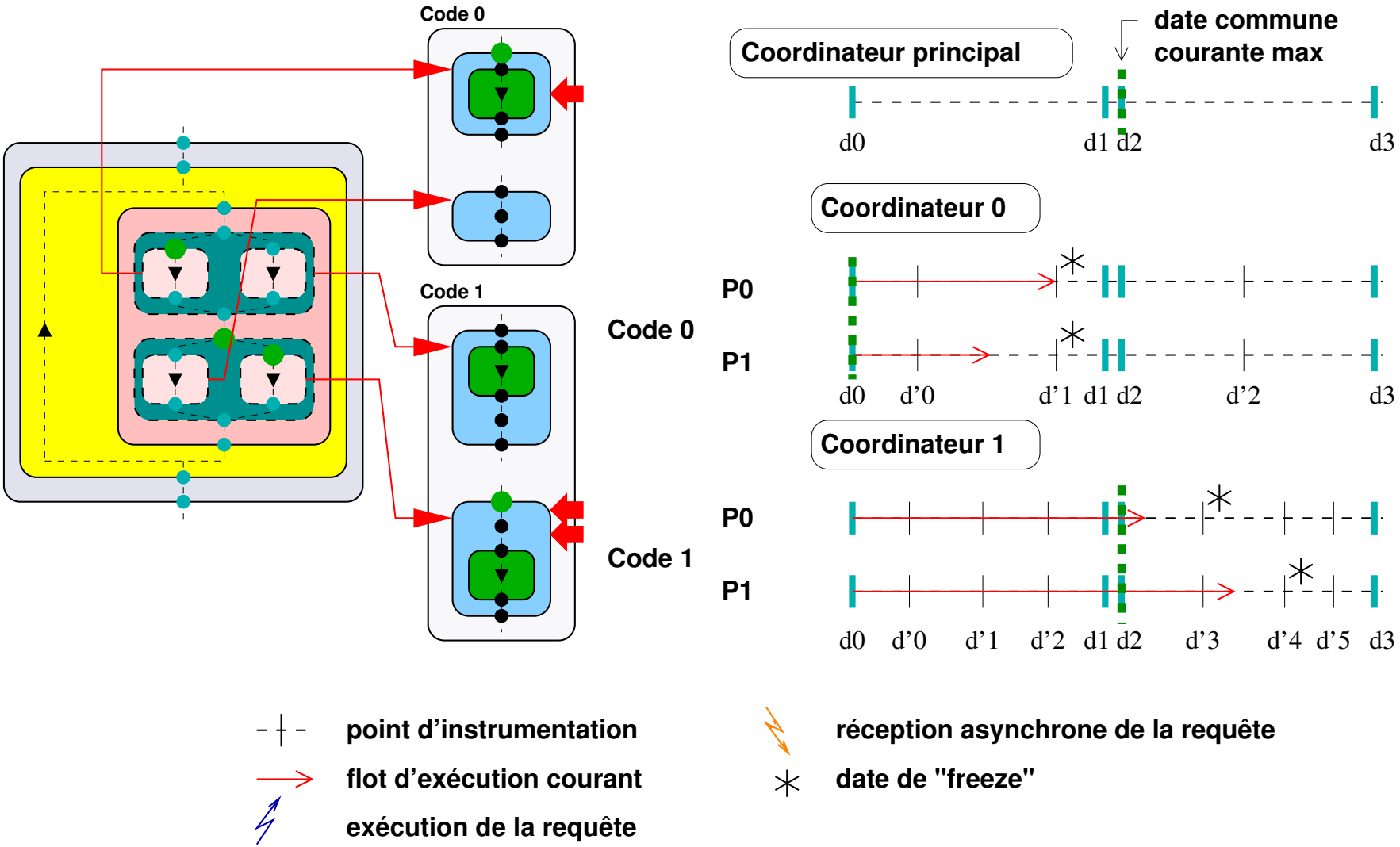
Planifier une date de traitement afin de garantir la cohérence temporelle

Réception d'une requête et "freeze" des points



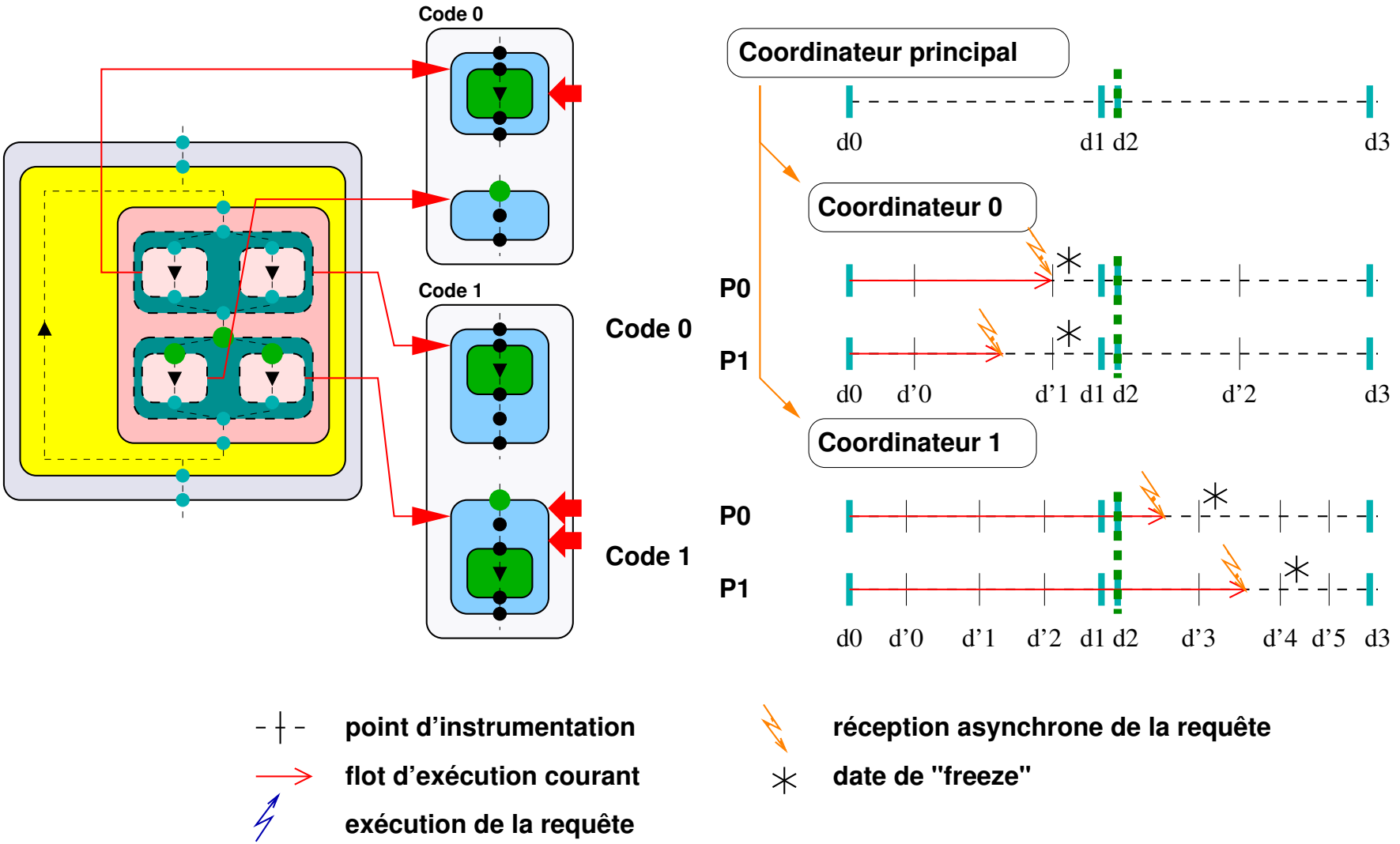
Planifier une date de traitement afin de garantir la cohérence temporelle

Détermination de la date commune courante max



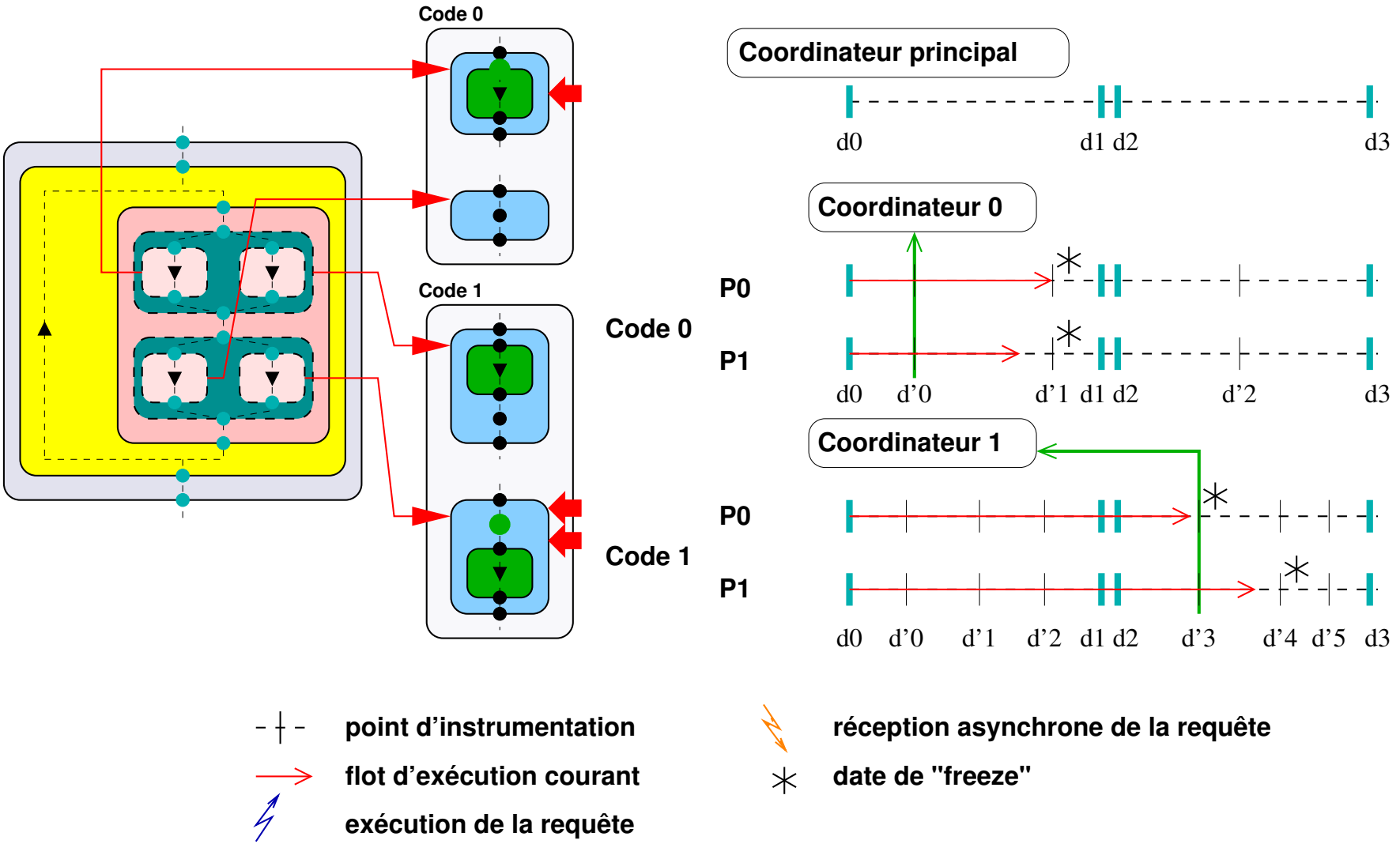
Planifier une date de traitement afin de garantir la cohérence temporelle

Envoi de la date commune courante max avec la requête



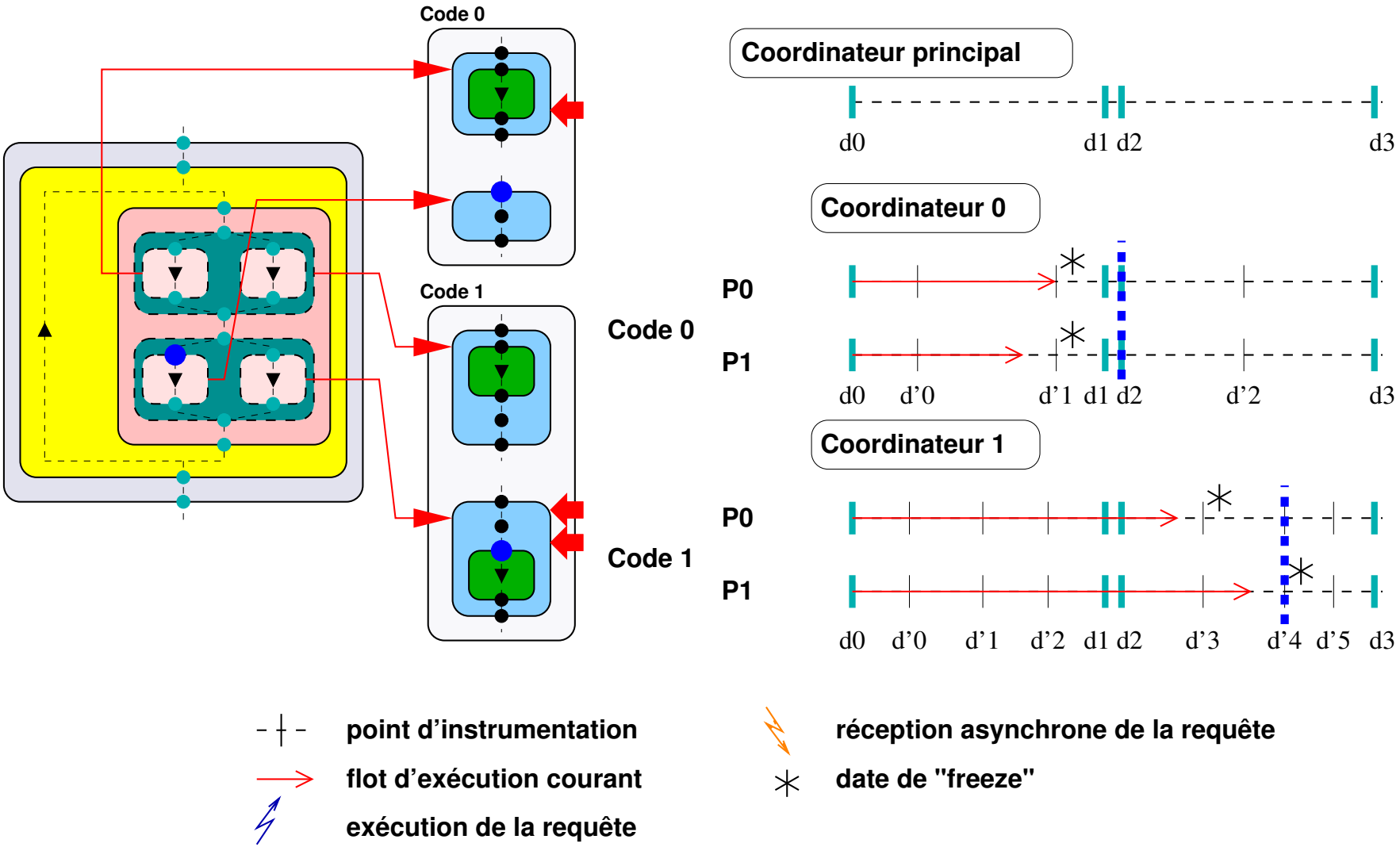
Planifier une date de traitement afin de garantir la cohérence temporelle

Détermination des dates courantes locales max



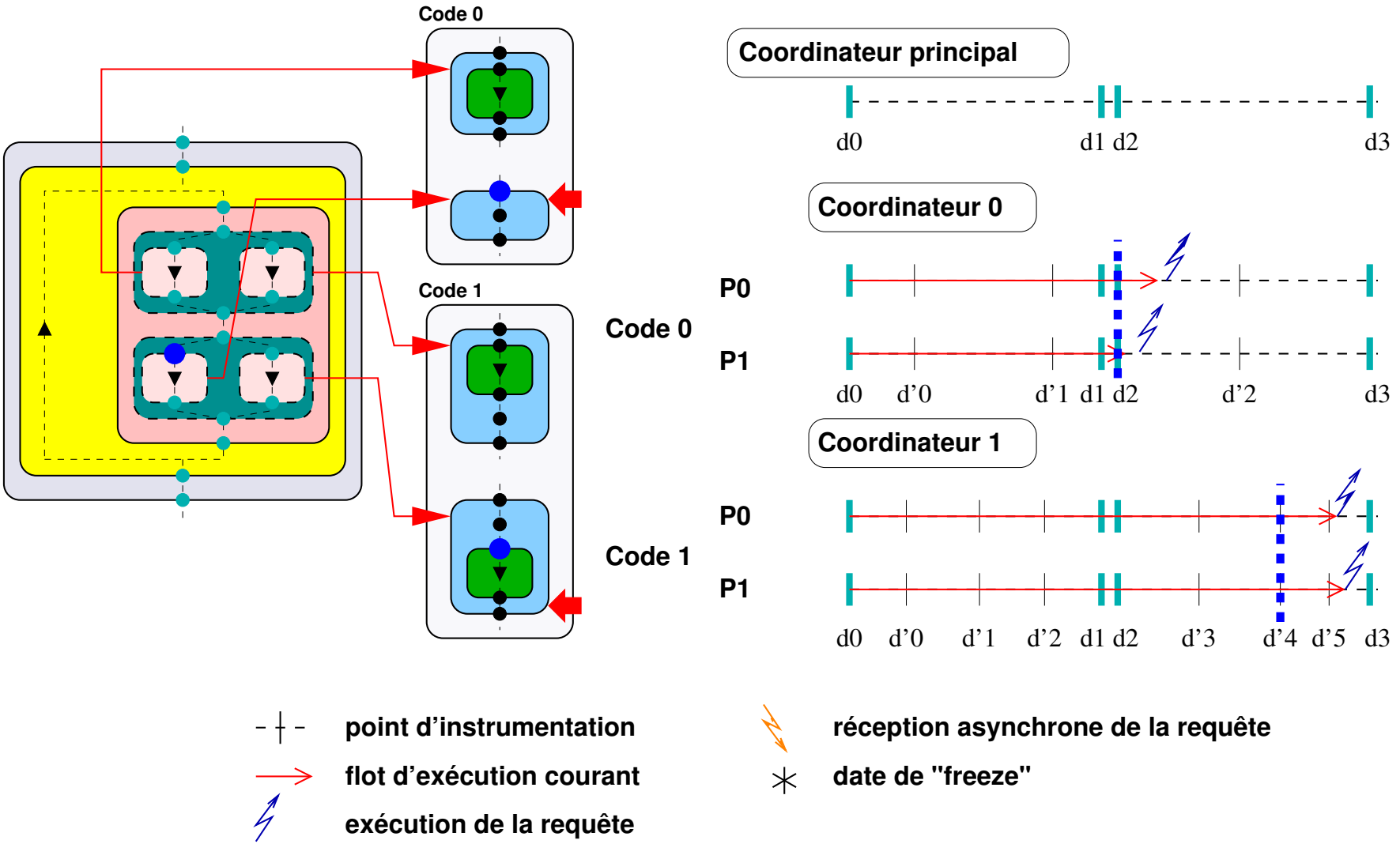
Planifier une date de traitement afin de garantir la cohérence temporelle

Détermination des dates de planification locales



Planifier une date de traitement afin de garantir la cohérence temporelle

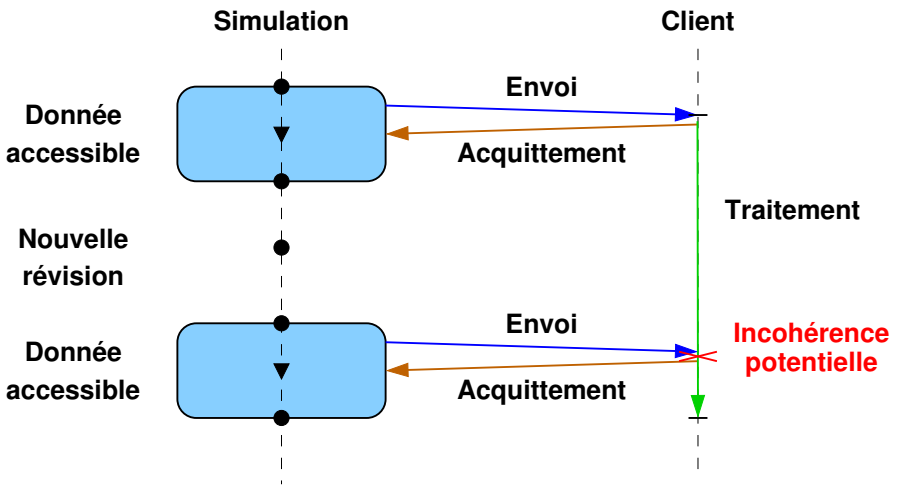
Exécution asynchrone des requêtes



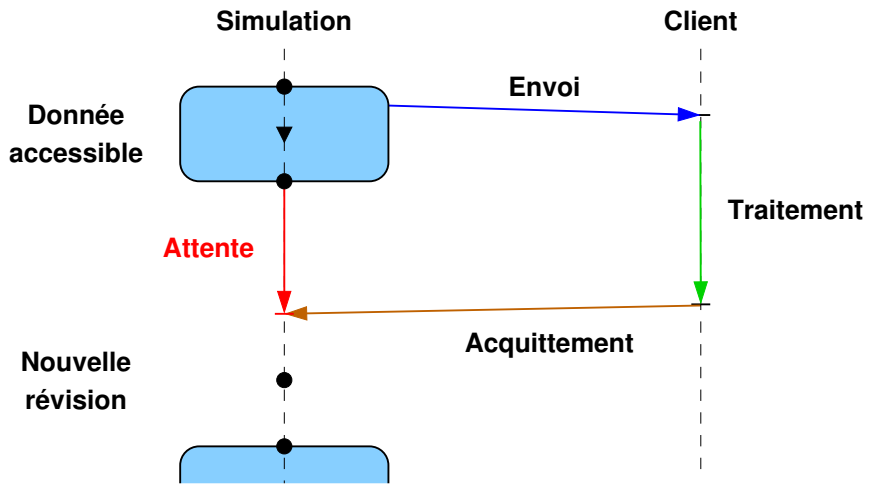
Cohérence coté client

S'assurer que les données restent cohérentes dans le client

- Problème potentiel avec les requêtes répétées
  → les données sont envoyées dès qu'une nouvelle version est produite
- Dans EPSN
  → **acquittement au plus tôt** : après les envois
  → **acquittement au plus tard** : après les traitements
- Dans EPSN2
  → **acquittement au plus tôt avec verrou**
  → après réception des données, prendre un verrou en écriture sur les données et le relâcher une fois les post-traitements finis



(a) Acquittement au plus tot

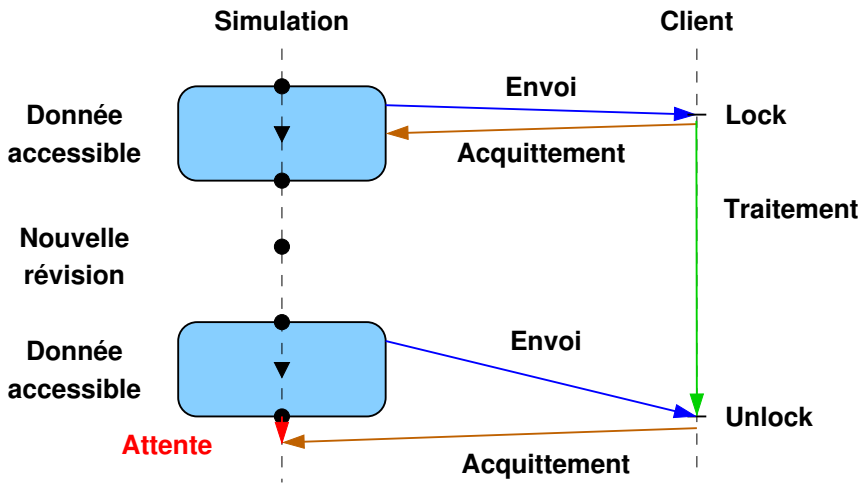


(b) Acquittement au plus tard

Cohérence coté client

S'assurer que les données restent cohérentes dans le client

- Problème potentiel avec les requêtes répétées
  → les données sont envoyées dès qu'une nouvelle version est produite
- Dans EPSN
  → **acquittement au plus tôt** : après les envois
  → **acquittement au plus tard** : après les traitements
- Dans EPSN2
  → **acquittement au plus tôt avec verrou**
  → après réception des données, prendre un verrou en écriture sur les données et le relâcher une fois les post-traitements finis



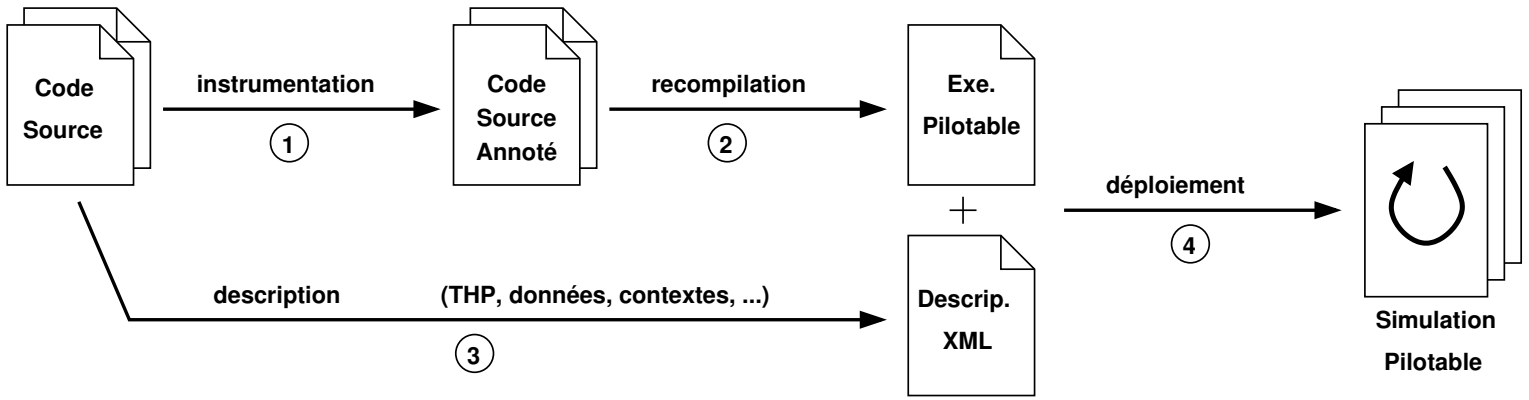
Acquittement au plus tot avec verrou

Plan

Principales fonctionnalités

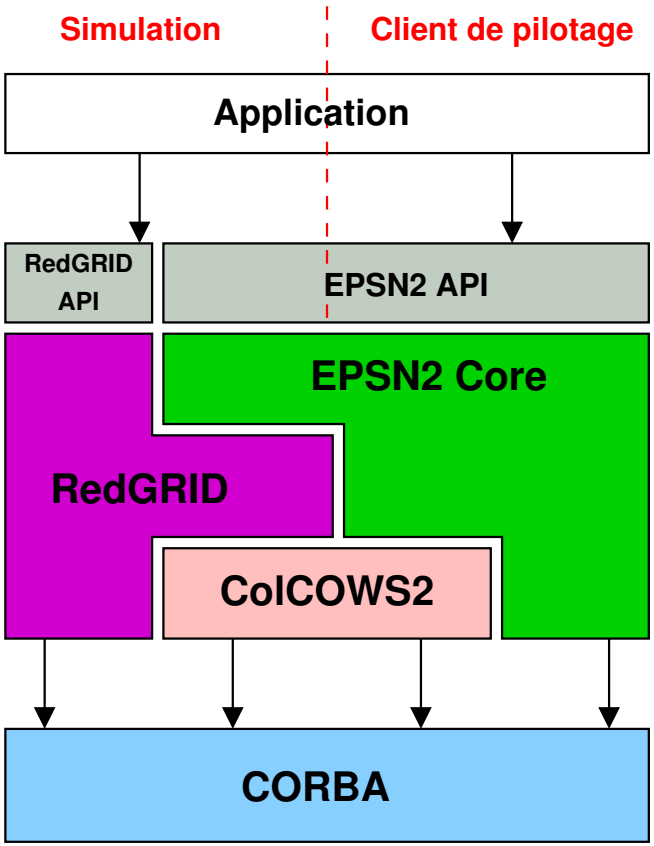
- Pilotage de simulations existantes
- Contrôle à distance du flot d'exécution
- Extraction de données pour la visualisation en ligne
- Modification des données à la volée
- Plate-forme distribuée et dynamique basée sur CORBA

Processus d'intégration et de déploiement



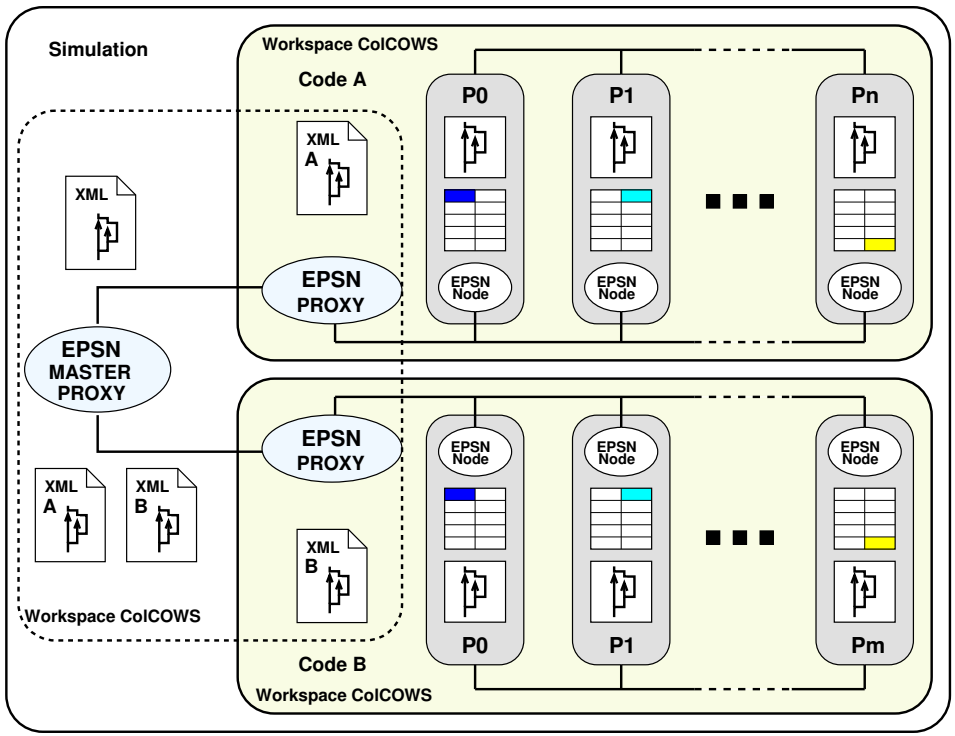
Couches logicielles de la plate-forme EPSN2

- ColCOWS2 : mise en commun des objets CORBA et de communications collectives CORBA
- RedGRID : couche de description et de transfert des données
- EPSN2 Core : gestionnaire de requêtes, coordination, *etc.*



Architecture de la plate-forme EPSN2

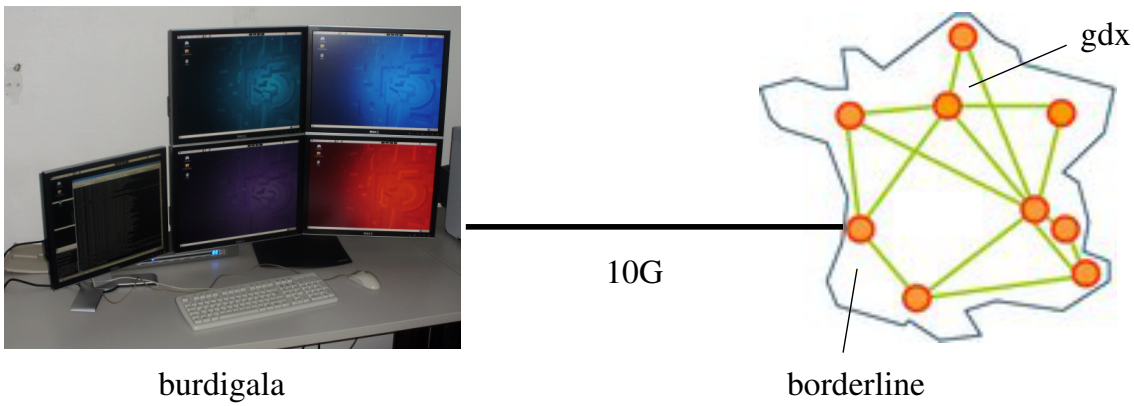
- Un espace de travail ColCOWS par code avec un proxy
- Un nœud par processus de la simulation
- Un espace de travail contenant les proxy avec un proxy principal



Plan

Conditions expérimentales

- Cluster de calcul Grid'5000 Bordeaux
  → *gdx* à Orsay : 126 bi-Opteron, réseau Giga-Ethernet
  → *borderline* à Bordeaux : 10 quadri-Opteron dual-core, réseau Myrinet/Infiniband
- Cluster de visualisation
  → 4 bi-Opteron, réseau Giga Ethernet/Infiniband, carte graphique nVidia Quadro FX 4500X2
- Logiciels
  → MPICH2/MVAPICH, Open MPI
  → OmniORB 4

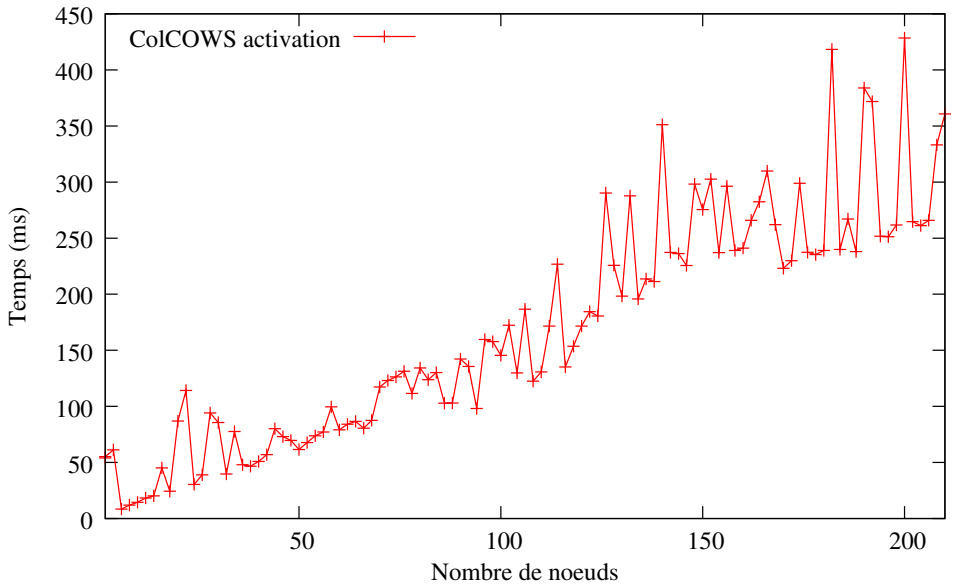


Mesures effectuées

- Mesures préliminaires sur la bibliothèque ColCOWS2
  → activation d'un espace de travail
  → communications collectives
- Mesures préliminaires sur le noyau d'EPSN2
  → temps d'initialisation de la plate-forme
  → temps de planification
  → recouvrement dans les clients de visualisation
- Mesures sur une simulation M-SPMD

Activation d'un espace de travail CoICOWS

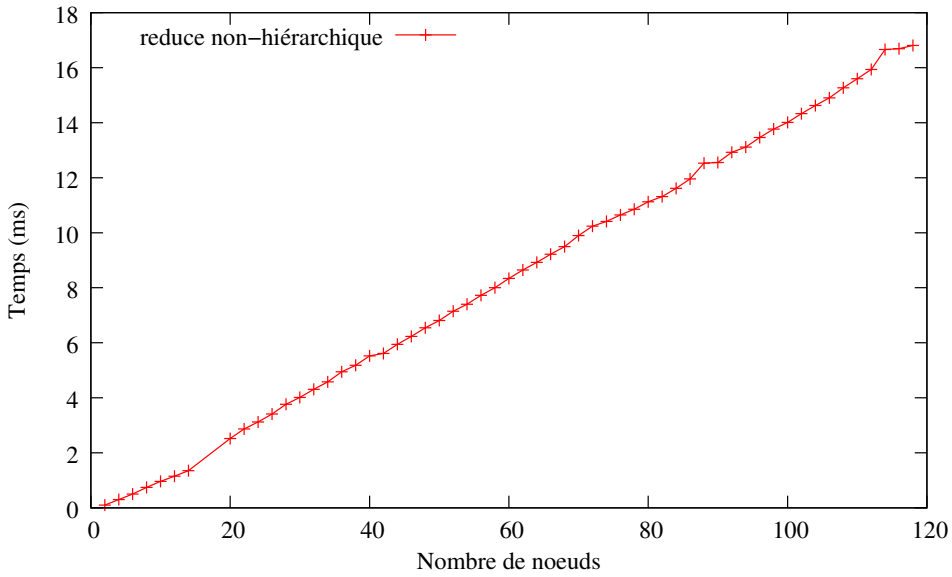
- Mise en commun de la connaissance des nœuds
- Effectuée lors de l'initialisation de la plate-forme EPSN



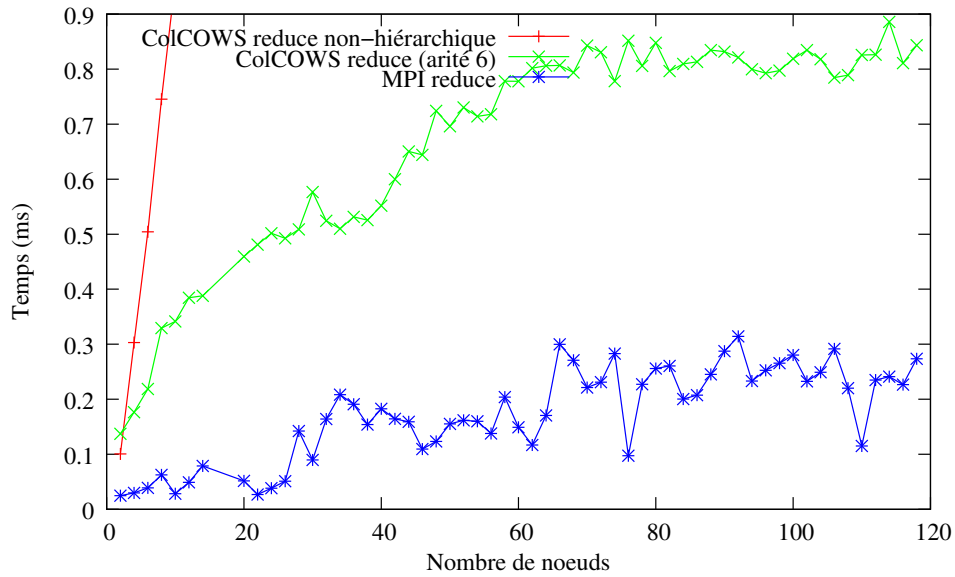
- Temps linéaire dû à la centralisation de l'information
- Aléas venant de la surcharge du service de nommage

Communications collectives de type reduce

Communications utilisées dans EPSN



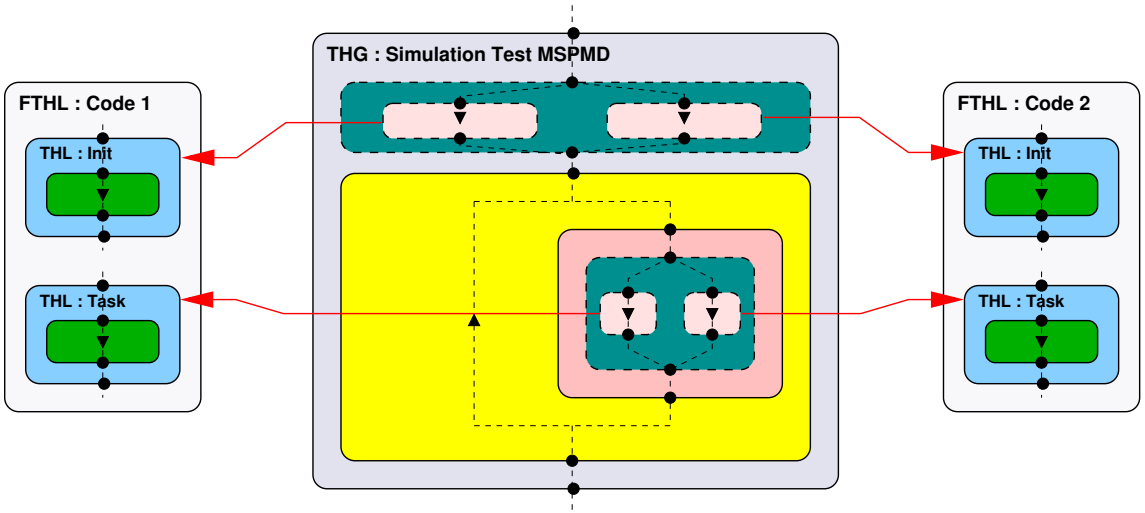
Communications utilisées dans EPSN2



→ **Gain d'un rapport 20**

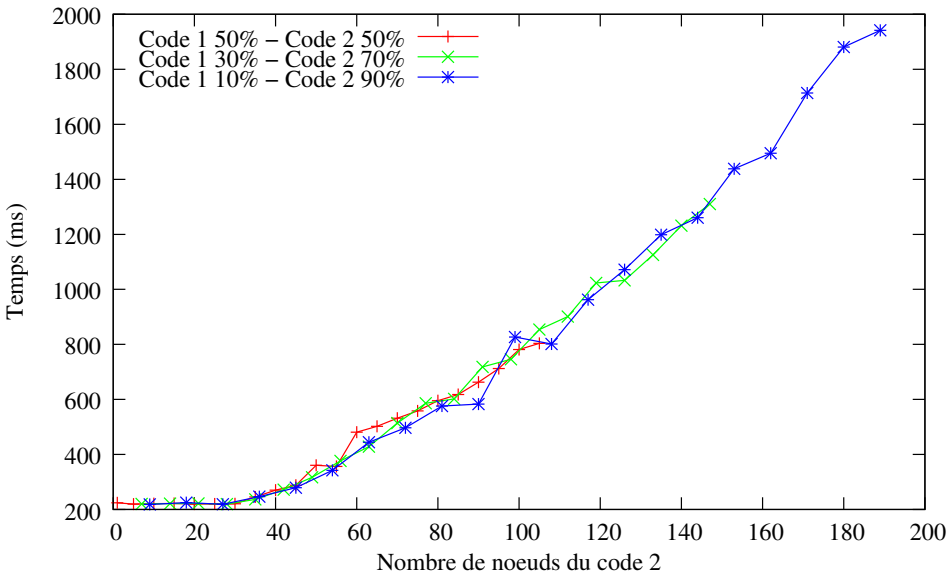
Présentation de la simulation de test

- Simulation M-SPMD
- Simulation paramétrable
  → nombre de codes couplés
  → durée de la tâche de calcul (THL : *Task*)



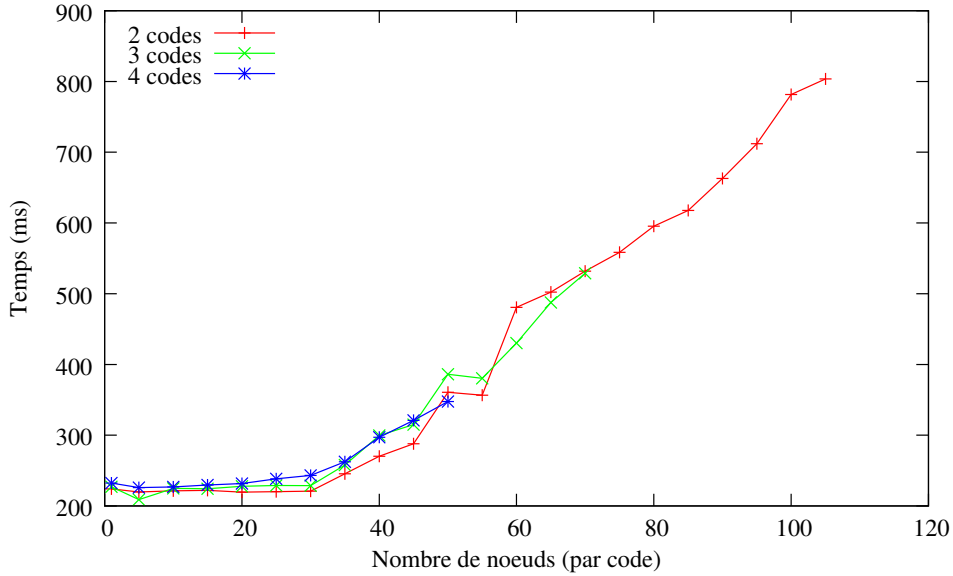
Temps d'initialisation de la plate-forme

Initialisation de la plate-forme pour 2 codes de simulation



→ **Limité par le nombre de nœuds maximum d'un code**

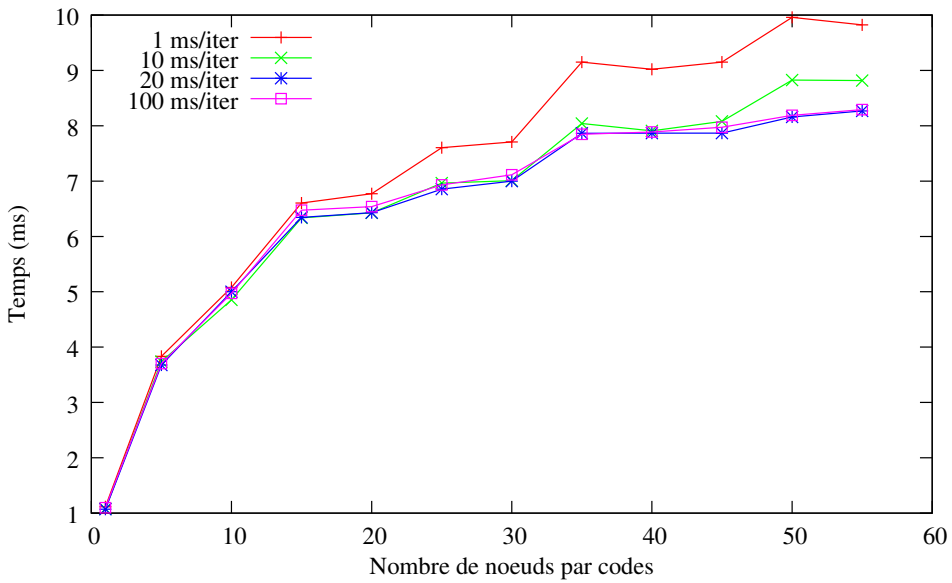
Initialisation de la plate-forme avec un nombre variable de codes



→ **Légèrement influencé par le nombre de codes**

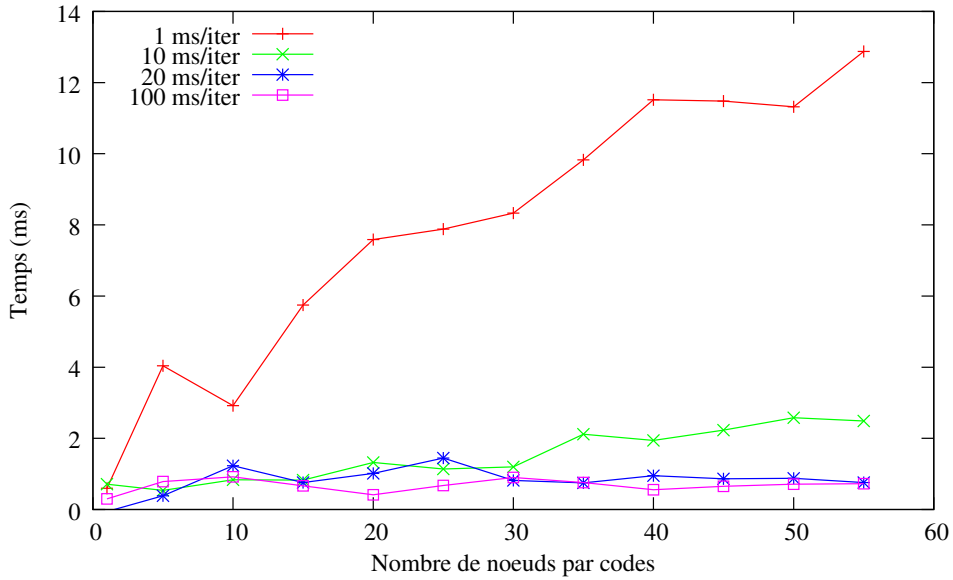
Temps de planification (500 requêtes "test" sur 1000 itérations)

Temps de planification



→ **Temps de planification idéal de** 8 *ms*

Surcoût de la planification



→ **Faible surcoût si temps de l'itération > temps de planification**

Mesure préliminaire visualisation

- Deux tâches : une tâche où la donnée est produite A, une où elle est accessible B
- Un client faisant 100 *ms* de post-traitement

	Configuration (en ms)			Mesures (en ms)		Surcoût total
Type acquittement	tc_A	tc_B	tp	tc_A	tc_B	sur $tc_A + tc_B$
Acquittement tard	1	1	100	1.02	101.32	99.34
Acquittement tôt	1	1	100	1.05	99.08	98.13
Acquittement tard	1	100	100	1.02	101.19	1.21
Acquittement tôt	1	100	100	1.03	100.05	0.08
Acquittement tard	100	1	100	99.95	101.38	100.33
Acquittement tôt	100	1	100	99.99	1.37	0.36

- Acquittement au plus tard $\rightarrow$ recouvrement total si $tc_B > tp$
- Acquittement au plus tôt $\rightarrow$ recouvrement total si $tc_A + tc_B > tp$

Propagation d'une onde dans un cristal d'Argon

→ 286 262 atomes, 26 130 éléments triangulaires et 13 347 sommets
→ simulation exécutée sur 8 processeurs : 4 pour MD et 4 pour FE
→ pas de temps à vide : 111.76 *ms* par itération
→ pas de temps avec instrumentation seule : 112.83 *ms* (0.9 %)

	Surcoût			
Période (nb d'itér.)	1		10	
Dumper ParaView	531.16 *ms*	375.3 %	154.05 *ms*	37.8 %
Transfert séquentiel (tard)	284.39 *ms*	154.5 %	124.18 *ms*	11.1 %
Transfert séquentiel (tôt)	274.32 *ms*	145.5 %	124.90 *ms*	11.8 %
Visu. séquentielle (tard)	450.82 *ms*	303.4 %	147.66 *ms*	32.12 %
Visu. séquentielle (tôt)	308.65 *ms*	176.2 %	125.06 *ms*	11.9 %
Visu // sur 4 nœuds (tôt)	242.08 *ms*	116.6 %	114.09 *ms*	2.1 %

Contact glissant de surfaces rugueuses

→ 1 033 124 atomes, 41 472 éléments tétraèdriques et 7 681 sommets
→ simulation exécutée sur 50 processeurs : 40 pour MD et 10 pour FE
→ pas de temps à vide : 469.05 *ms* par itération
→ pas de temps avec instrumentation seule : 471.31 *ms* (0.5 %)

	Surcoût			
Période (nb d'itérations)	1		10	
Dumper ParaView	1 220.70 *ms*	160.2 %	542.93 *ms*	15.8 %
Transfert séquentiel (tard)	888.10 *ms*	89.3 %	519.97 *ms*	10.9 %
Transfert séquentiel (tôt)	864.79 *ms*	84.4 %	516.10 *ms*	10.0 %
Visu. séquentielle (tard)	2 857.14 *ms*	509.1 %	756.91 *ms*	61.4 %
Visu. séquentielle (tôt)	1 200.73 *ms*	156.0 %	516.98 *ms*	10.2 %
Transfert parallèle (tard)	698.77 *ms*	49.0 %	494.14 *ms*	5.3 %
Transfert parallèle (tôt)	687.58 *ms*	46.5 %	490.26 *ms*	4.5 %
Visu. parallèle (tard)	1 631.56 *ms*	247.8 %	623.19 *ms*	32.9 %
Visu. parallèle (tôt)	761.40 *ms*	62.3 %	494.20 *ms*	5.3 %

Conclusion

- Modélisation des simulations couplées de type Clients/Serveurs et M-SPMD
  → modèle unique pour toutes les simulations visées et *a priori* générique
- Définition de la cohérence pour les traitements de pilotage
  → définition de la cohérence en fonction des traitements de pilotage
  → introduction des algorithmes assurant la cohérence de bout en bout
- Conception et réalisation de la plate-forme EPSN2
  → prise en compte complète des simulations M-SPMD
  → prise en compte partielle des simulations Clients/Serveurs
- Mesures des performances et validation sur une “vraie” simulation
  → traitement de pilotage recouvert et cohérence assurée
  → performances obtenues essentiellement avec les communications collectives

Perspectives

Perspectives pour la plate-forme

- Prise en compte totale des simulations Client/Serveur
  → développements en cours dans l'ANR NOSSI
- Amélioration de la phase d'initialisation
  → regarder une version distribuée du service de nommage CORBA
- Prise en compte des simulations parallèles hybrides multi-processus/multi-threads
- Ajout de fonctionnalités d'aide au développement
  → aide au *checkpointing*
  → outils de *profiling*

Perspectives pour la modélisation

- Prise en compte des simulations paramétriques
  → simulation de type maître/esclaves
  → pilotage du maître pour contrôler l'espace des paramètres à tester
  → pilotage des esclaves en définissant la cohérence des données entre elles
- Enrichissement des contextes des tâches
  → ajouter des informations sur les tâches effectuant des communications
  → contextes permettant de spécifier quand on peut sauvegarder les données afin de faire du *checkpointing*