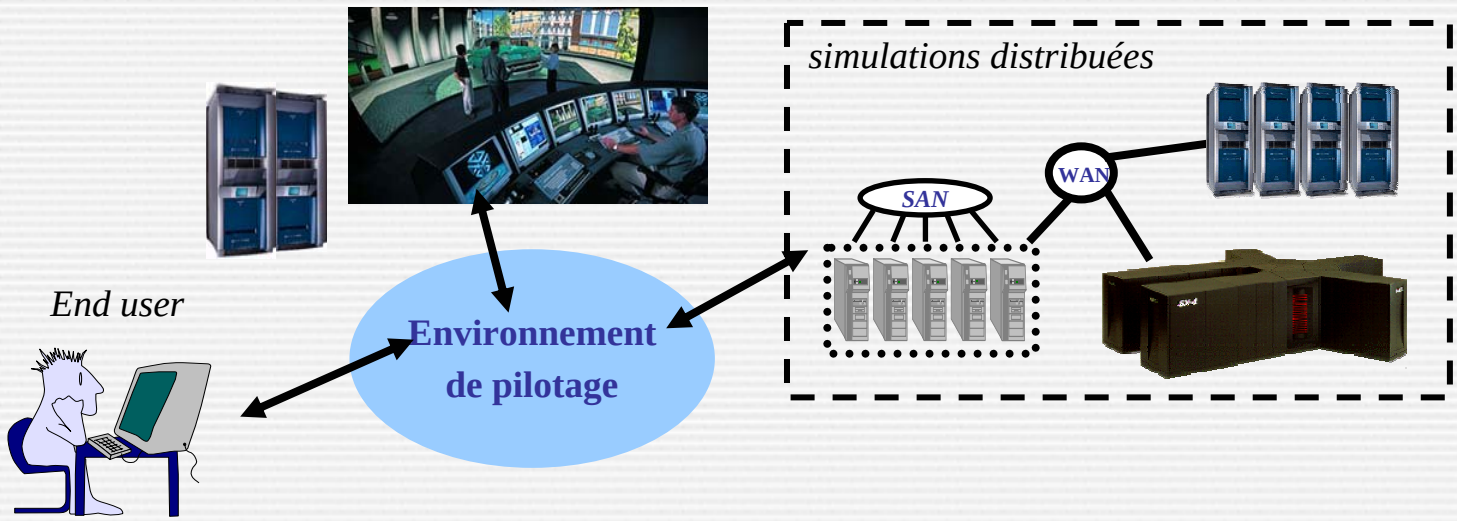
Pilotage de Simulations Numériques parallèles

L'environnement EPSN

Projet ScAlApplix - Inria Futurs Bordeaux /LaBRI/MAB

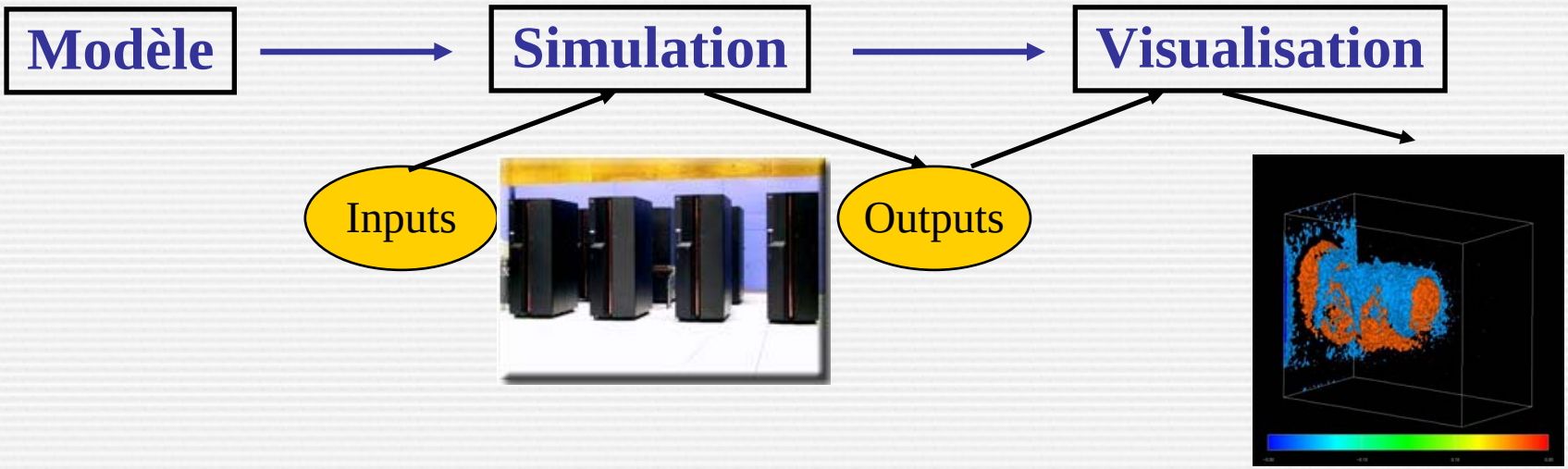
O. Coulaud , M. Dussère, A. Esnard



Plan

- « Computational Steering »
  - Le pilotage pourquoi ?
  - Les difficultés dans le cadre parallèle
  - État de l'art
- L'environnement EPSN
  - Modèle
  - Plate forme
  - EPSN et vision parallèle
- Un exemple pas à pas

Processus de simulation en calcul scientifique



mais ….

- Mode en batch,
- Pas d'interaction de l'utilisateur

Contexte

- Des simulations de plus en plus complexes
  - Amélioration des modèles physiques, mathématiques, …
  - Gros volumes de données (Terabyte, …)
- Augmentation des moyens de calcul
  - Processeurs, mémoires, machines parallèles,
  - réseaux rapides, grilles de calcul
- Côté visualisation
  - Augmentation des moyens
    - Des hardwares très performants, cluster graphique,
    - Diffusion du matériel R&V (Workbench, mur d'écran, …)
  - Des outils programmables de plus en plus performants :

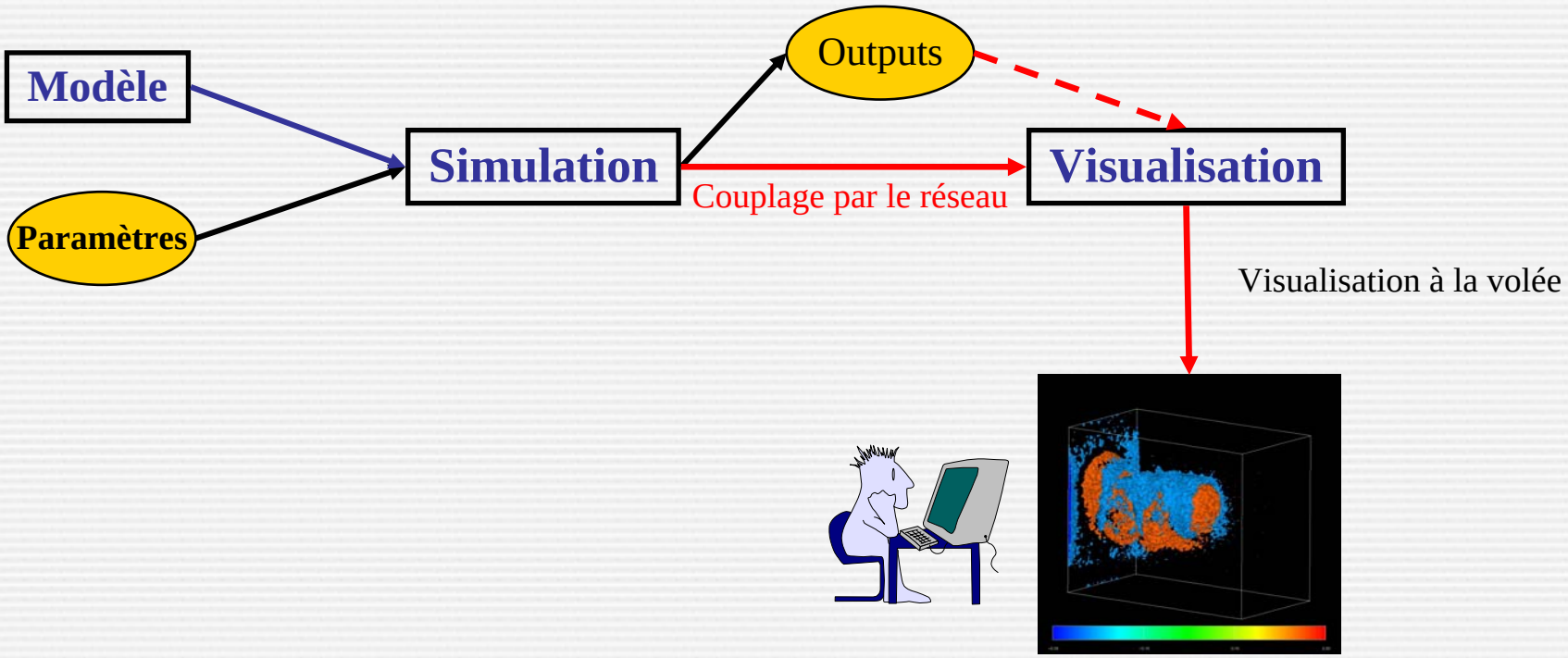
    AMIRA, AVS, VTK, Paraview …

    Visualisation parallèle et rendu parallèle

La visualisation en ligne

De nouveaux besoins apparaissent : suivre la simulation

Monitoring

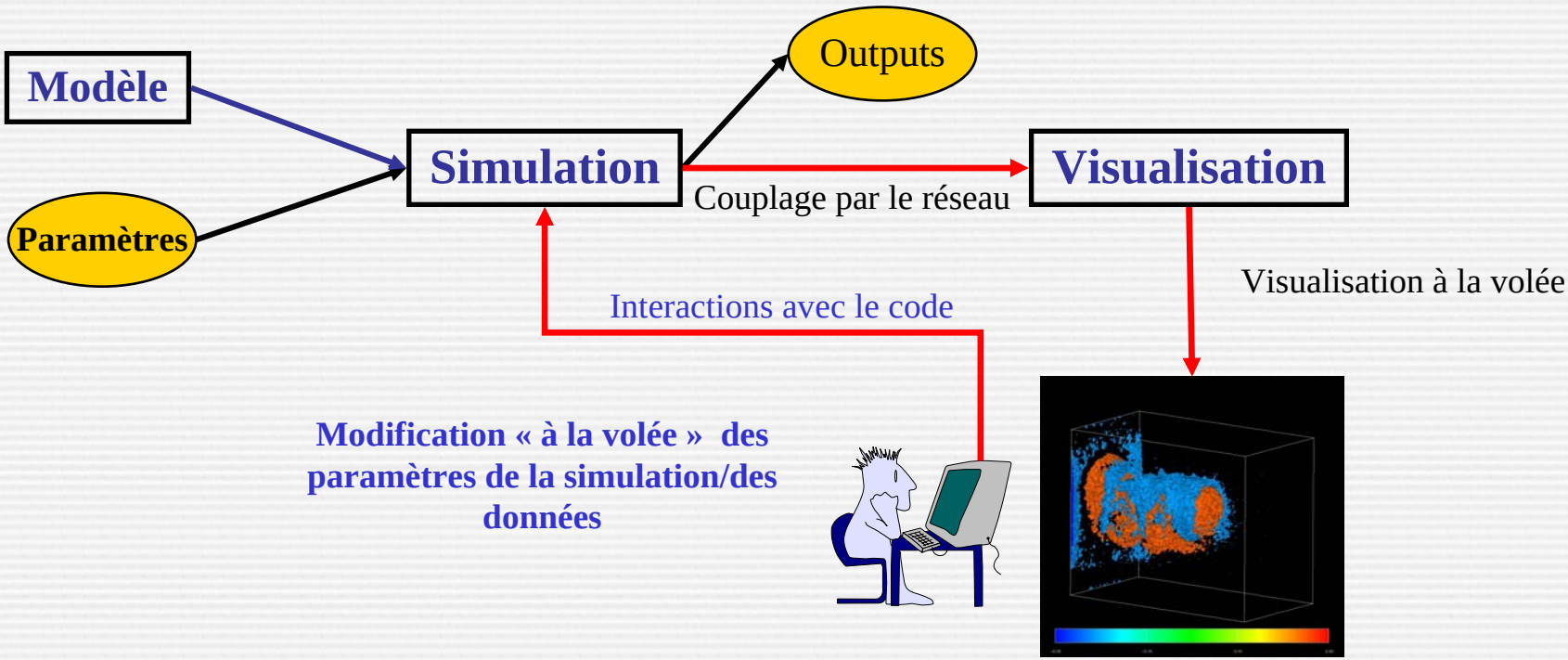


Observation passive du modèle

Simulation interactive

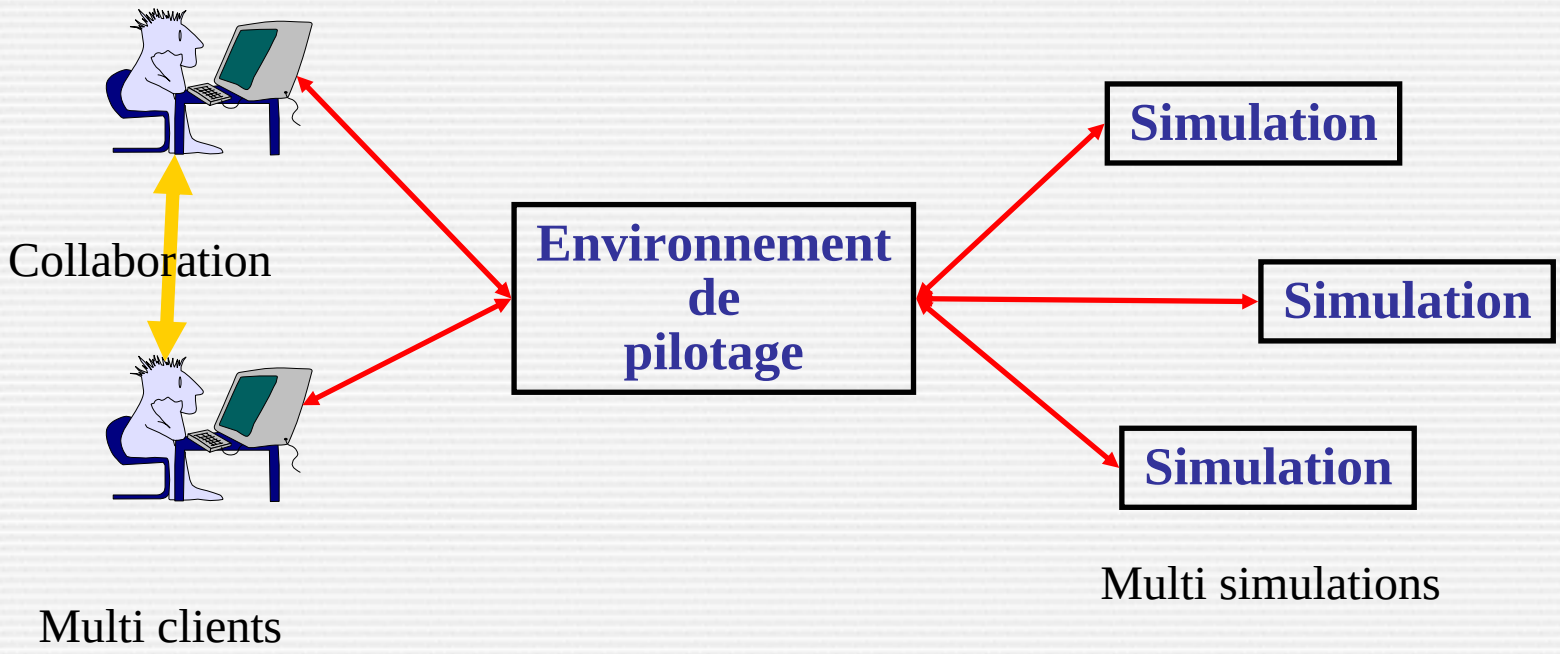
De nouveaux besoins apparaissent : interagir avec la simulation

Steering ou Pilotage



Passer du mode batch au mode interactif

Le mode collaboratif



Passer du mode interactif au mode collaboratif

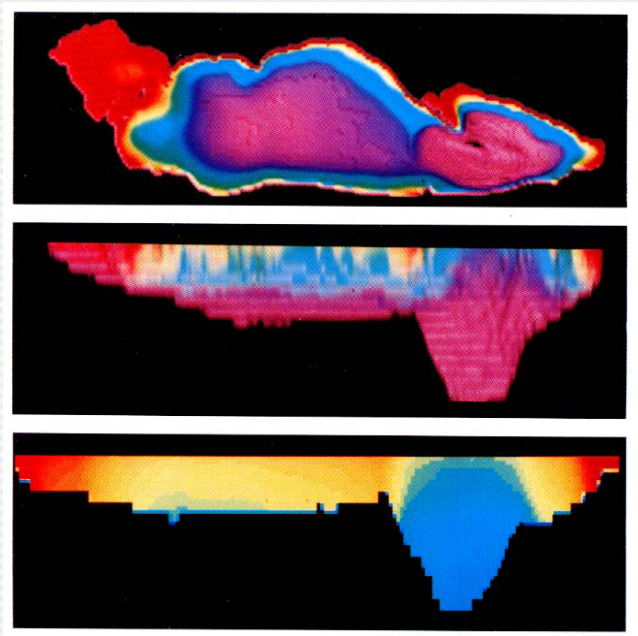
Computational Steering : pourquoi ?

- Efficacité
  - Éviter le gaspillage de temps CPU
    - Paramètres d'entrées erronés
    - Paramètres non pertinents
    - Instabilités numériques
  - Diminuer le temps entre le calcul et l'interprétation
- Se rapprocher du processus expérimental
  - Navigation dans l'espace des paramètres
  - Analyse de sensibilité
  - Analyse de cause à effet
- …

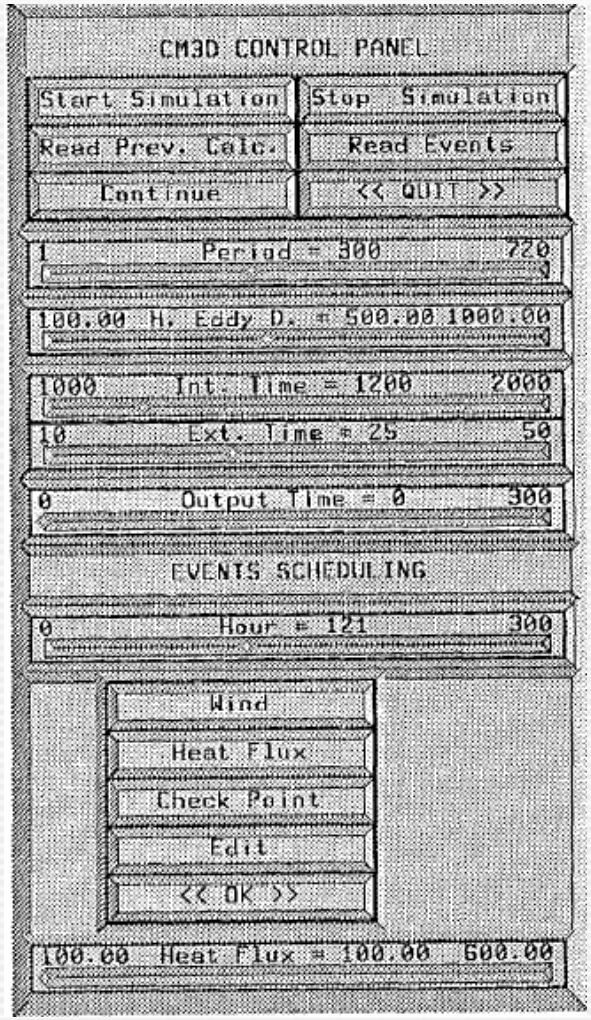
Quelques exemples de pilotage

Mécanique des fluides

- Turbulence 3D du lac Erie (1990)
  - Sterring
    - Conditions aux limites (vitesse du vent)
    - Événements (tempête avec flux de chaleur)



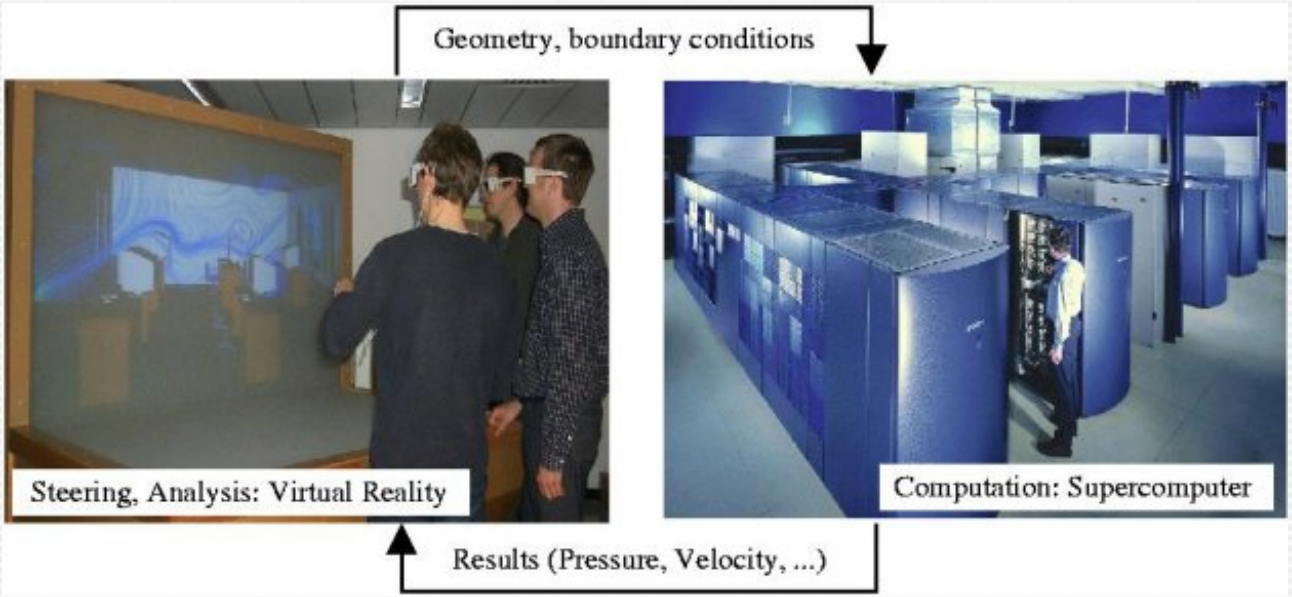
  - Échange en fin de pas temps
  - Grille de points



Marshall et all Ohio Supercomputer Center

Mécanique des fluides

- VirtualFluids_Real (2000) : Lattice-Bolzmann
  - Modification de la géométrie, des conditions aux limites
  - CFD en MPI et couplage avec PACX/MPI, échange en fin de pas temps
  - Holobench + OpenIventor



http://www.inf.bauwesen.tu-muenchen.de/personen/siggi/CompSteering/index.html

Biologie moléculaire

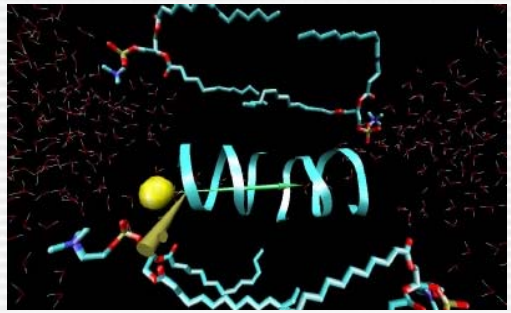
Dynamique moléculaire interactive - IMD

Couplage entre NAMD (DM parallèle) et VMD (visu séquentiel) via des sockets.

Échange en fin d'un pas de temps.

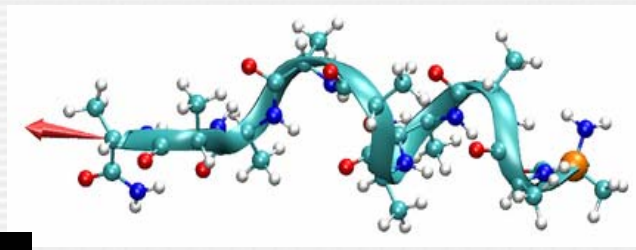
- Déplacement d'un atome

  Pull a sodium ion (large sphere) through the gramacidin A channel.



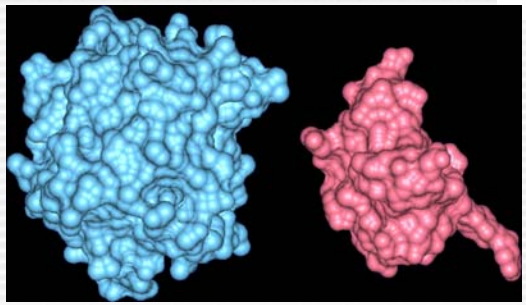
- Étirement d'une molécule

Stretching Deca-Alanine



- Docking

  Retour d'effort



Le Pilotage d’application

Computational Steering : utilisation

Trois niveaux d'utilisation :

1. Exploration du modèle, de l'espace des paramètres

   Interaction sur les données, géométrie, les conditions aux limites, …

2. Expérimentation d'algorithmes

   Changer dynamiquement de schémas de résolution, deboggueur

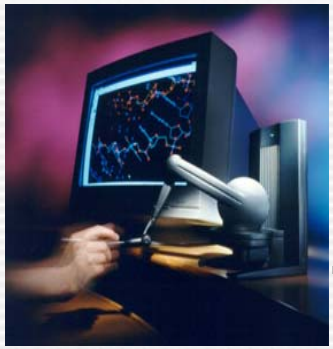
3. Analyse et optimisation des performances

   Interaction sur la configuration : équilibrage de charge, …

Agir sur les « données » de la simulation.

Computational Steering : plate-forme

Environnement de pilotage

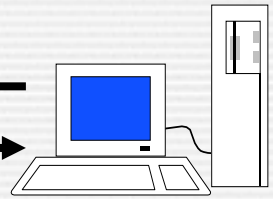


Visualisation et Pilotage

Simulation Numérique

interprétation

interactions



commandes

résultats

Utilisateur

Interface utilisateur

Système de communication

Simulation

Couplage de codes :

- simulation et visualisation
- interactivité sur un des codes

Computational Steering : les composants

Une architecture de CS est constituée de 3 composants

1. La simulation numérique

   Intégrer une simulation (instrumentation du code : actions, données, …)

2. Le système de communication
   - Extraire les données de la simulation
   - Diffuser les actions de pilotage

3. Le pilotage

   Interagit sur la simulation à partir
   - de la visualisation (humaine)
   - de l'algorithme (automatique)
   - d'un mode hybride

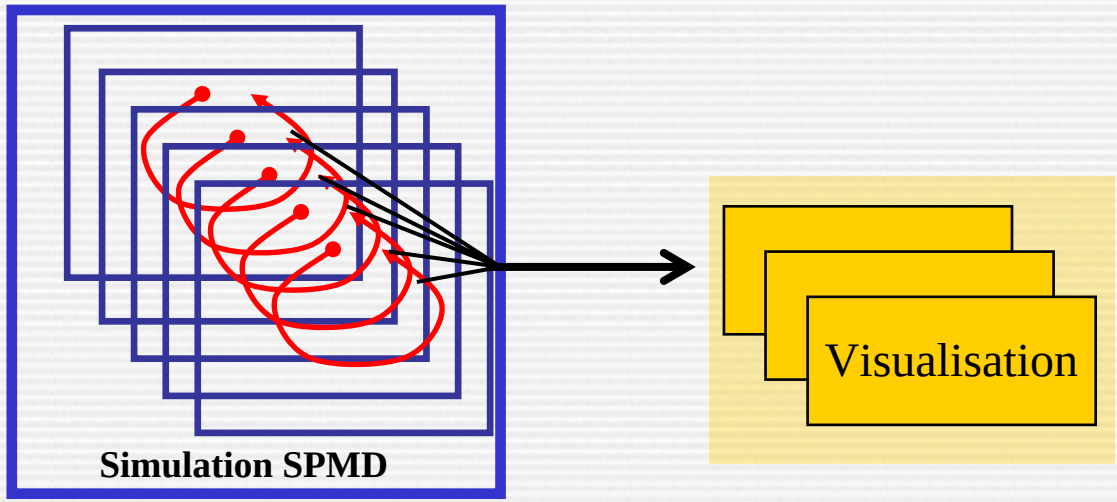
Computational Steering : caractéristiques

1. Intégrer facilement une simulation numérique
   Annotation des sources
   Encapsulation dans un module – approche PSE

2. Interagir
   - Temps réel
   - Interaction cohérente : conserver la sémantique du modèle physique
   - Interaction efficace

3. fonctionnalités
   - Dynamique connecter, déconnecter – approche client-serveur
   - Gérer plusieurs simulations, plusieurs clients

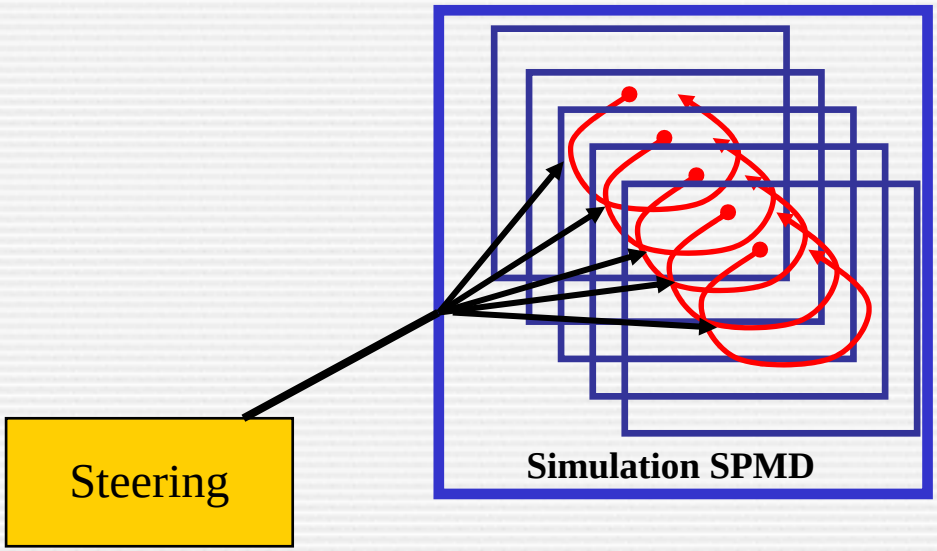
Simulations parallèles et steering : les difficultés



Difficultés

- Extraire les données en assurant une cohérence
  - locale/spatiale : point ou plage d'accès de la donnes (consistance des données)
  - globale/temporelle : même itération entre tous les processeurs
- Performance

Simulations parallèles et steering : les difficultés



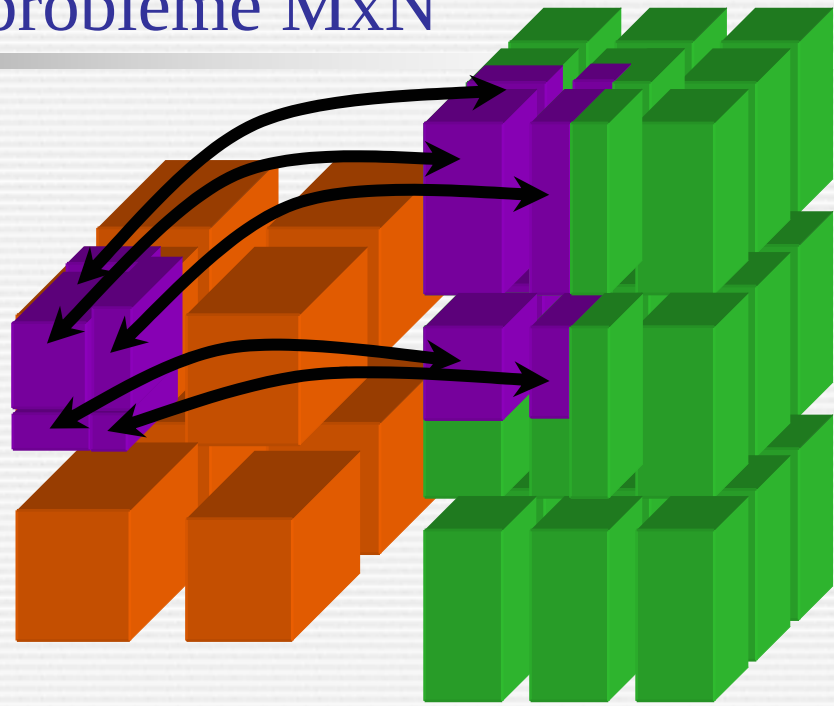
Difficultés

- Assurer que l'action ait lieu en même temps
- Assurer la cohérence spatiale du traitement ie au même endroit

Simulations parallèles et steering : les solutions

- Modéliser la simulation
  - Vision du code
    - Vision plate avec un ou plusieurs points d'instrumentation
    - Boucle de calcul
    - Graphe de tâches (arête = flot d'exécution)
  - Vision des données
    - Type (maillage, particules, …) et nature (graphique, pilotable)
    - Distribution
    - droit d'accès, …
- Performance
  - Recouvrir communications calculs
    - Communications asynchrones, plages d'accès ou bufferisation
  - Flux parallèles

Transfert de données ou le problème MxN



Différentes approches

1. Centralisation des données sur le maître
   - Communication point à point
   - Diffusion des données
2. Flux parallèle
   - Données en place
   - Traitements parallèles

Problème de la redistribution

- Description des données (field, maillage, particules)
- Description de distribution entre processeur et le stockage mémoire
- Bibliothèque de communication

État de l’art

État de l'art

- Des environnements pour interagir avec la simulation

Applications

CUMULVS	CFD /généraliste	PVM+AVS puis CCA et MxN
(OViD) Viper	mécanique des fluides	deboggueur // + visualisation
Magellan	physique	langage de pilotage interprété
VASE	généraliste	control flow + data –flow
VMD+NAMD	dynamique moléculaire	API (socket)

Images

SCIRun	médecine, fluides, géophysique	PSE
CSE	généraliste	Client/Serveur de données
COVISE	généraliste	PSE+Collaboratif
VISIT	DM, géophysique	API + AVS/Express
CAVEStudy	généraliste	CAVERNSoft, orienté R&V
….		

Grid

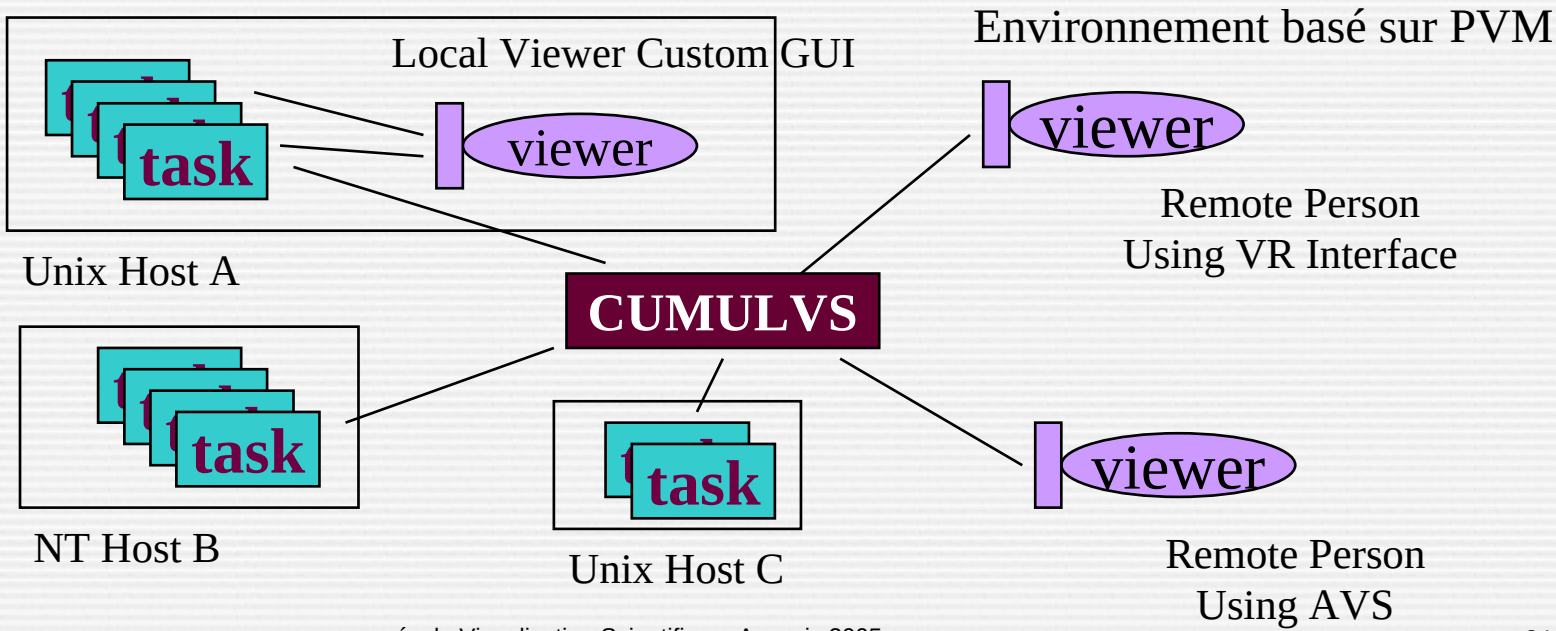
CACTUS	Onde gravitationnel/ généraliste	XML, HTTP,HDF5,
RealityGRID	Généraliste	Web services
DISCOVER	Généraliste	Navigateur web
….		

Collaborative, User Migration, User Library for Visualization and Steering

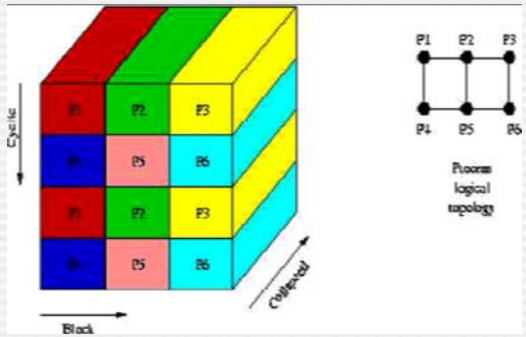
Infrastructure collaborative pour interagir avec des simulations scientifiques

Caractéristiques

- Client / serveur (simulation)
- Plusieurs simulations parallèles, plusieurs clients séquentiels
- Collaboration



Collaborative, User Migration, User Library for Visualization and Steering



Données :

- Field avec une distribution bloc cyclique (HPF)
- Particules

Points d'instrumentation

- Un unique point pour envoyer les données à la simulation
  (communications asynchrones)
- Points de reprises (tolérance aux fautes)

Applications

combustion, mécanique des fluides, sismique

Projet actif

- Réalité virtuelle – CAVE (performer + VTK)
- CCA : Common Component Architecture (www.acl.lanl.gov/cca-forum)

http://www.csm.ornl.gov/cs/cumulvs.html

CSE : Computational Steering Environment

Domaine de R&V

Modèle : client / serveur /client

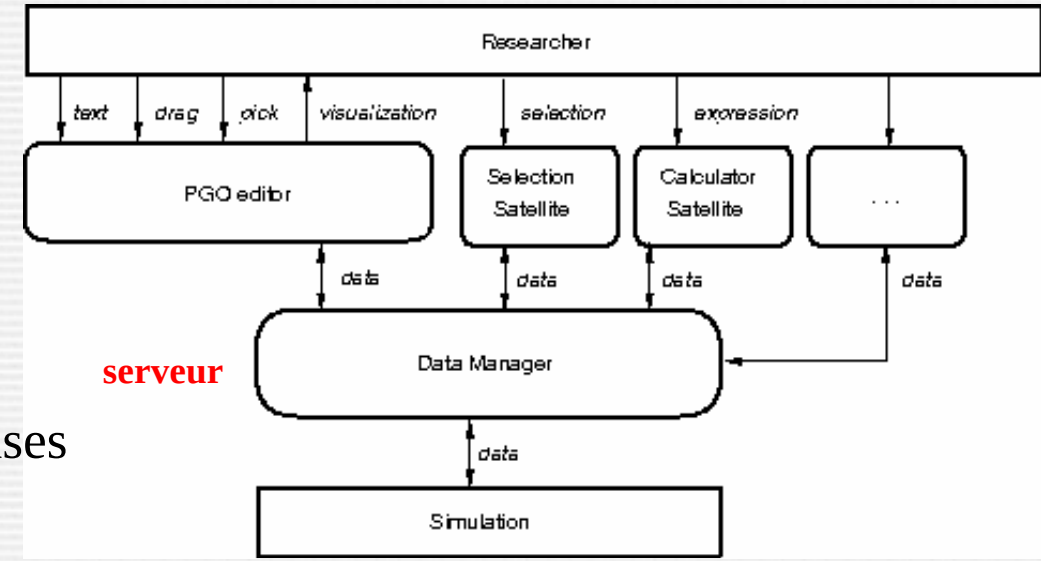
Multi clients et multi simulations

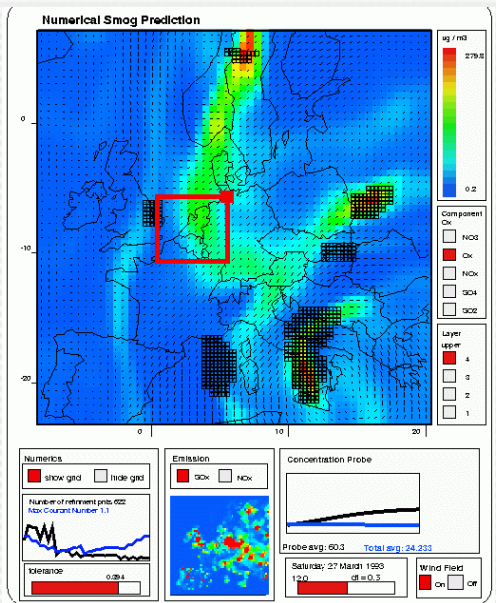
Données : scalaires, tableaux denses

Approche avec une base de données

Éditeur graphique d'interface de
pilotage mode édition/exécution

http://www.cwi.nl/projects/cse





Bilan

- Beaucoup d'environnements mais
  - Domaine scientifique spécifique (fluides, géophysique, médecine, …)
  - Monde de la réalité virtuelle (collaboratif, …) ou du Grid computing

- Caractéristiques
  - Interaction avec l'application

    annotation des sources + API (init, breakpoint, …)

    intégration de la simulation dans un module
  - Applications séquentielles ou parallèles
  - Communications

    serveur centralisé, protocole de bas niveau

Peu d'environnements s'intéressent à la fois

- à obtenir un couplage performant
- à intégrer une simulation existante
- à la visualisation performante et/ou immersive.

Visualisation et simulations

- La visualisation post-traitement
- La visualisation en ligne
  - Regarder en temps réel les données de la simulation
- L'interaction : modifier le comportement de la simulation
  - Modifier des variables sans toucher les structures de données
  - Sauvegarde, rejouer la simulation, …
  - Modifier un algorithme : remaillage local, …

Nécessite souvent de restructurer le code

A l'intersection de plusieurs disciplines

- Problèmes issus du couplage
  - Couplage
  - Déploiement de l'application, gestion des ressources
  - Dynamique connexion/déconnexion à la volée
  - Communications efficaces entre les codes

} Grid

- Interaction
  - Exploration : extraction, repérage, navigation
  - Pilotage collaboratif : centralisé, réparti
  - IHM

} Visualisation
Réalité Virtuelle
Science cognitive

- Simulation
  - L'algorithmique du couplage, …
  - Exploration des paramètres, ….

} Applicatif

L'environnement EPSN

Michael Dussere

INRIA - Institut National de Recherche en Informatique et Automatisme
unité de recherche Futurs Bordeaux

26th April 2005

Approche impérative

L'approche impérative est basée sur des actions prédéfinies.

- Dirigée par la simulation
- Commandes de steering dans le code la simulation
- Préparation explicite de messages et envois systématiques

Pros & Cons

- Très peu flexible
- Toute modification nécessite une nouvelle instrumentation
- Instrumentation simple

Approche impérative

L'approche impérative est basée sur des actions prédéfinies.

- Dirigée par la simulation
- Commandes de steering dans le code la simulation
- Préparation explicite de messages et envois systématiques

Pros & Cons

- Très peu flexible
- Toute modification nécessite une nouvelle instrumentation
- Instrumentation simple

Approche descriptive

L'approche descriptive est basée sur les propriété de la simulations qui sont révélée par l'instrumentation.

- Dirigée par le client
- Définition des données d'interaction
- Définition de *droits d'accès* dans l'instrumentation

Pros & Cons

- Flexible
- Toute modification nécessite une nouvelle instrumentation
- L'instrumentation devient très complexe quand on prend en compte des simulations parallèles

Approche descriptive

L'approche descriptive est basée sur les propriété de la simulations qui sont révélée par l'instrumentation.

- Dirigée par le client
- Définition des données d'interaction
- Définition de *droits d'accès* dans l'instrumentation

Pros & Cons

- Flexible
- Toute modification nécessite une nouvelle instrumentation
- L'instrumentation devient très complexe quand on prend en compte des simulations parallèles

Approche générique

L'approche générique utilisée dans EPSN est une approche descriptive où une partie de la description est dissociée de l'instrumentation

- Dissociation du squelette et des autres informations
- Description externe combinant le squelette et les autres informations
- Description du squelette de la simulation dans l'instrumentation

Pros & Cons

- Flexible
- Les modifications de la description externe ne nécessitent pas une nouvelle instrumentation
- L'instrumentation reste simple
- La généricité permet la gestion du parallélisme et d'opération de haut niveau

Approche générique

L'approche générique utilisée dans EPSN est une approche descriptive où une partie de la description est dissociée de l'instrumentation

- Dissociation du squelette et des autres informations
- Description externe combinant le squelette et les autres informations
- Description du squelette de la simulation dans l'instrumentation

Pros & Cons

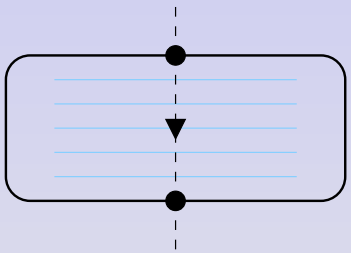
- Flexible
- Les modifications de la description externe ne nécessitent pas une nouvelle instrumentation
- L'instrumentation reste simple
- La généricité permet la gestion du parallélisme et d'opération de haut niveau

Plan

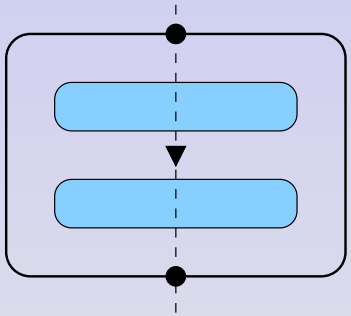
Graphe de Taches Hiérarchiques - HTG

HTG

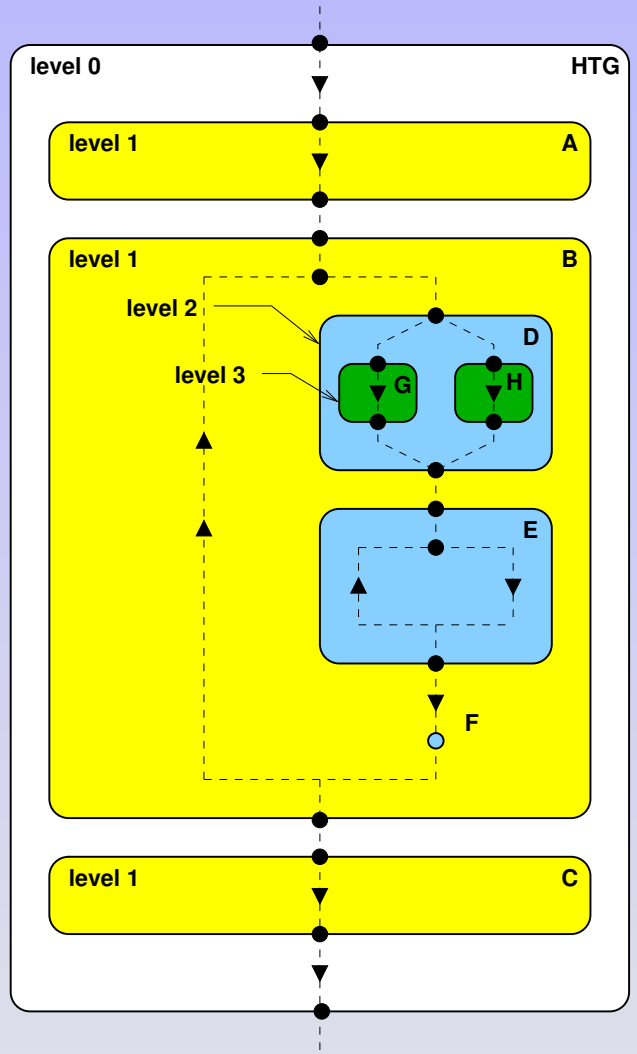
- Graphe dirigé acyclique
- Les noeuds encapsulent du code
- Les noeuds peuvent contenir un HTG
- Les arêtes suivent le flux de contrôle



tâche simple



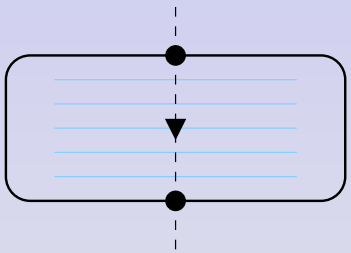
tâche composée



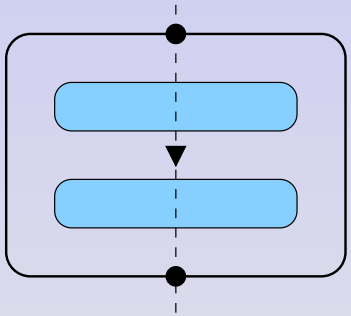
Graphe de Taches Hiérarchiques - HTG

HTG

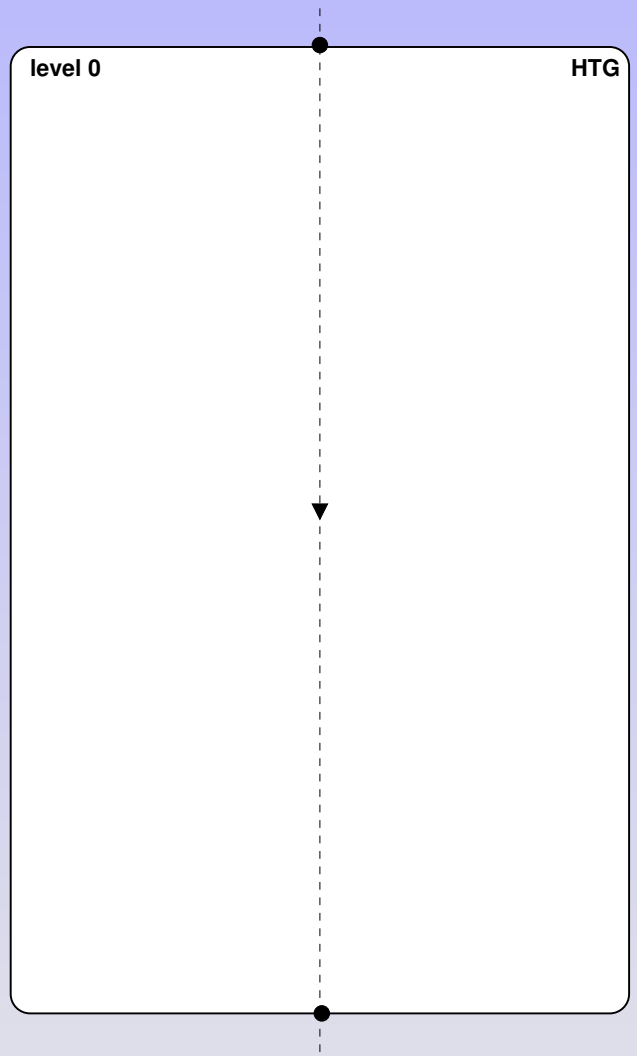
- Graphe dirigé acyclique
- Les noeuds encapsulent du code
- Les noeuds peuvent contenir un HTG
- Les arêtes suivent le flux de contrôle



tâche simple



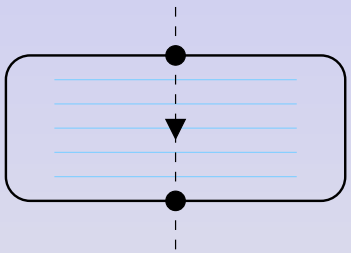
tâche composée



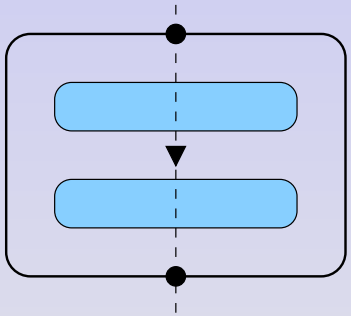
Graphe de Taches Hiérarchiques - HTG

HTG

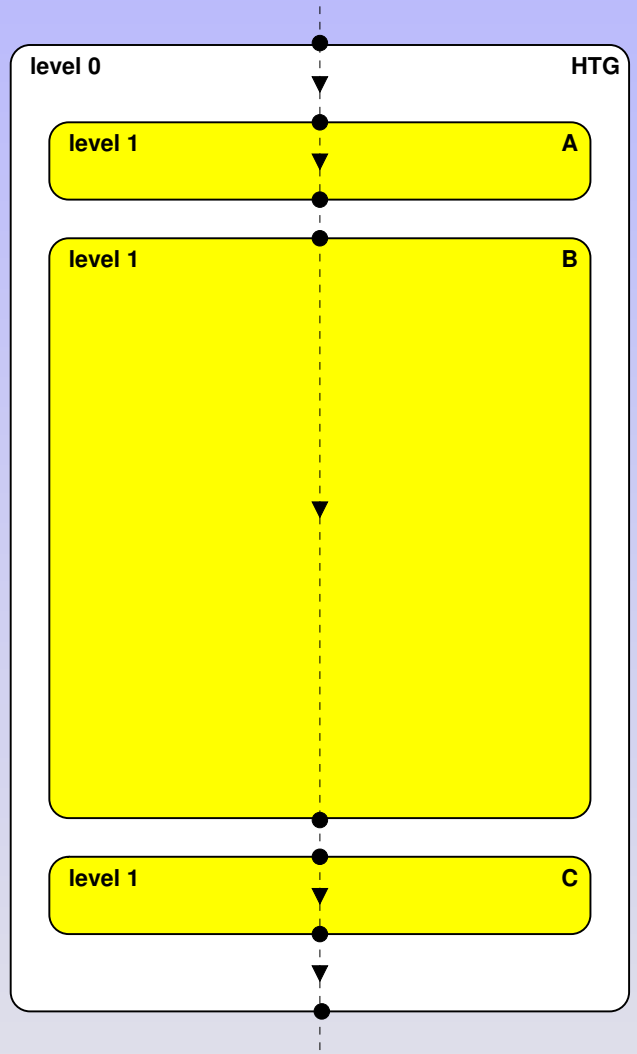
- Graphe dirigé acyclique
- Les noeuds encapsulent du code
- Les noeuds peuvent contenir un HTG
- Les arêtes suivent le flux de contrôle



tâche simple



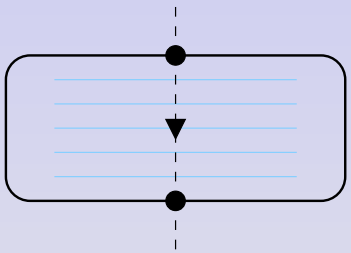
tâche composée



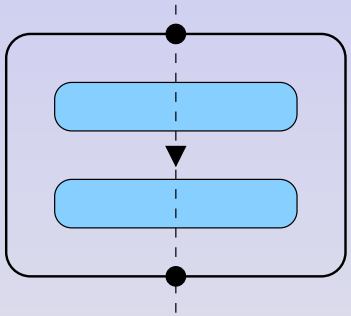
Graphe de Taches Hiérarchiques - HTG

HTG

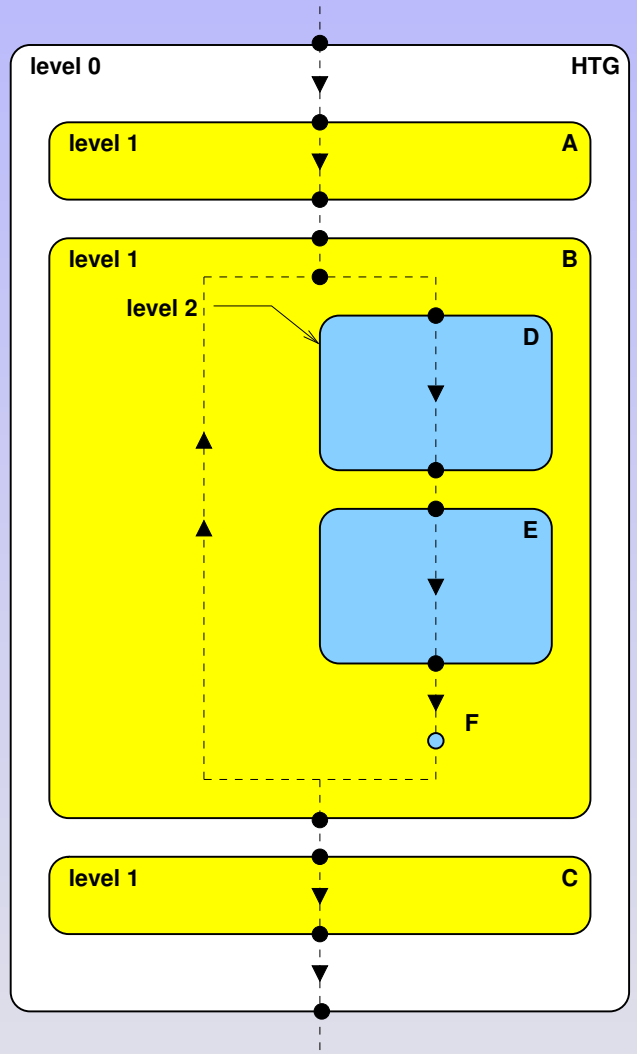
- Graphe dirigé acyclique
- Les noeuds encapsulent du code
- Les noeuds peuvent contenir un HTG
- Les arêtes suivent le flux de contrôle



tâche simple



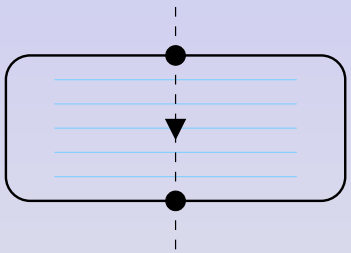
tâche composée



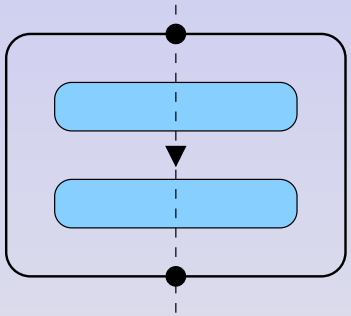
Graphe de Taches Hiérarchiques - HTG

HTG

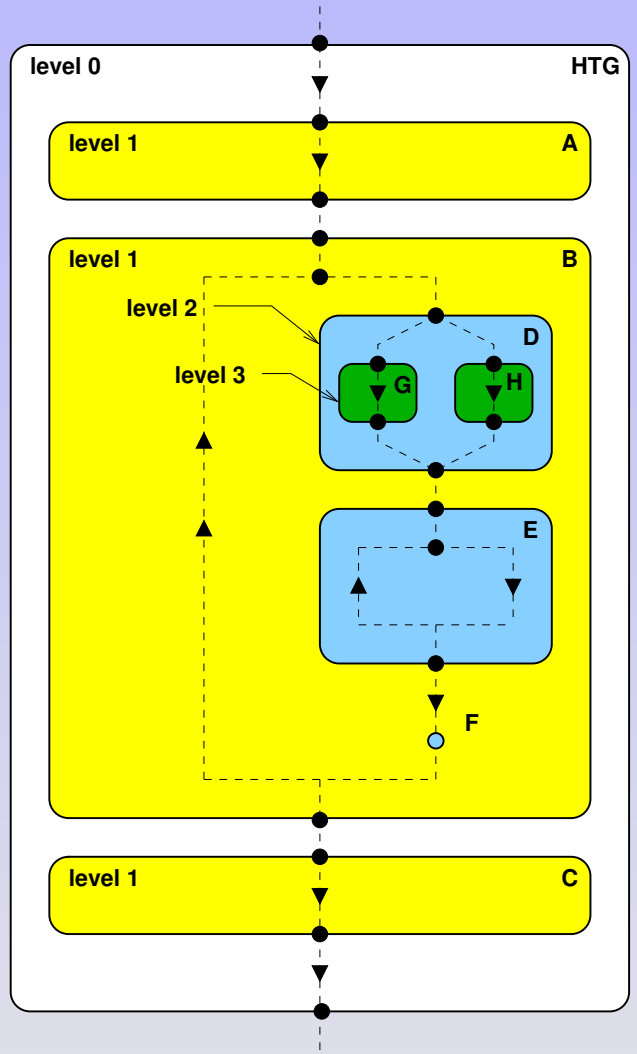
- Graphe dirigé acyclique
- Les noeuds encapsulent du code
- Les noeuds peuvent contenir un HTG
- Les arêtes suivent le flux de contrôle



tâche simple



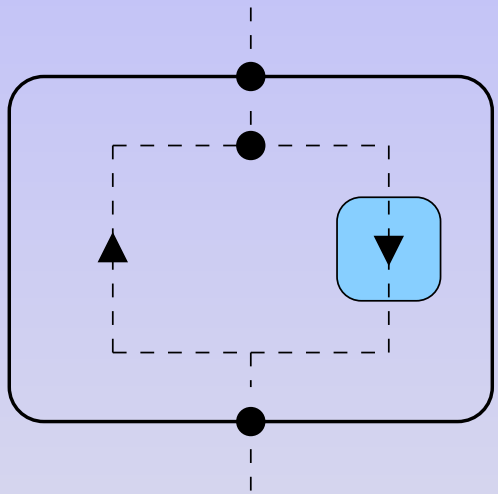
tâche composée



Les taches en boucles

Propriétés

- Tache hiérarchique contenant une seule sous-tache
- La sous-tache peut être parcouru plusieurs fois
- Garantit le caractère acyclique du graphe
- Modélise les structures itératives: `do`, `for`, `while`
- Ne peut modéliser les renvois type `goto`

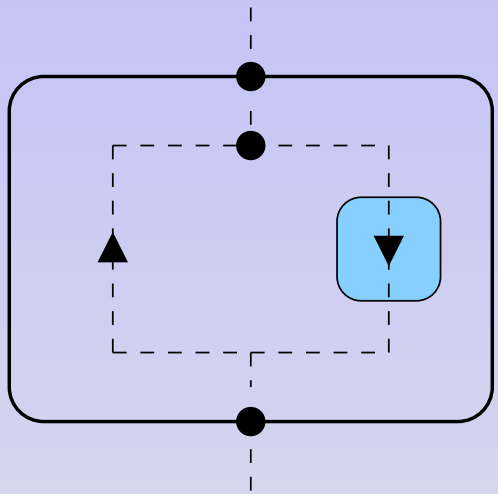


tâche en boucle

Les taches en boucles

Propriétés

- Tache hiérarchique contenant une seule sous-tache
- La sous-tache peut être parcouru plusieurs fois
- Garantit le caractère acyclique du graphe
- Modélise les structures itératives: `do`, `for`, `while`
- Ne peut modéliser les renvois type `goto`

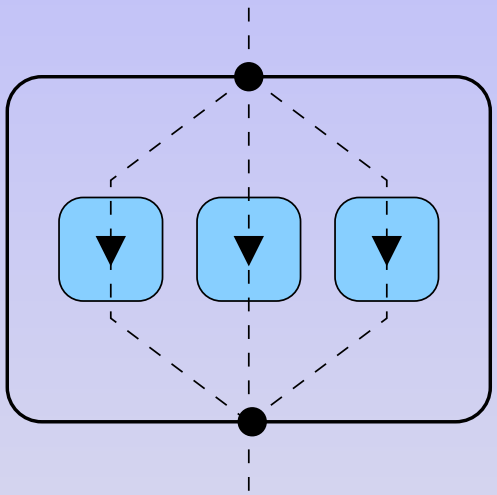


tâche en boucle

Les switch

Propriétés

- Tache hiérarchique contenant une ou plusieurs sous-taches
- Les sous-taches s'excluent mutuellement
- Modélise les structures conditionnelles: `if/then/else`, `switch/case`



tâche conditionnelle

Le HTG dans les simulations parallèles

Principe

- Tous les processus suivent le même HTG
- les points d'instrumentation ne sont pas synchronisants
- La simulation est dans un état cohérent si :
  - tous les processus sont sur le même point du HTG
  - (pour une boucle) tous les processus sont sur la même itération
  - (pour un switch) tous les processus sont sur la même branche
  - les parents de la tache courante présentent la même cohérence.

Propriétés pour le parallélisme

- **acyclique**
  $\Rightarrow$ *datation* précise de l'évolution des processus
- **hiérarchique**
  $\Rightarrow$ robustesse face à des différences d'exécution mineures

Le HTG dans les simulations parallèles

Principe

- Tous les processus suivent le même HTG
- les points d'instrumentation ne sont pas synchronisants
- La simulation est dans un état cohérent si :
  - tous les processus sont sur le même point du HTG
  - (pour une boucle) tous les processus sont sur la même itération
  - (pour un switch) tous les processus sont sur la même branche
  - les parents de la tache courante présentent la même cohérence.

Propriétés pour le parallélisme

- **acyclique**
  $\Rightarrow$ *datation* précise de l'évolution des processus
- **hiérarchique**
  $\Rightarrow$ robustesse face à des différences d'exécution mineures

La description des données

Description minimale

- Identifiant:
  lien avec la bibliothèque de redistribution
- Classe:
  scalaire, champs, particules, maillage ...
- Localisation:
  - localisé sur un processus
  - répliqué sur tous les processus
  - partagé entre les processus

Plan

Pilotage par les requêtes

Principe

- La simulation joue le rôle de serveur en attente de requêtes clientes
- La simulation peut être connectée à plusieurs clients
- Les interactions sont des réponses à des requêtes client
- Les communications et les synchronisations sont des réponses aux requêtes client
- Pas de requêtes $\Rightarrow$ Pas de sur-coût

Pilotage par les requêtes

Principe

- La simulation joue le rôle de serveur en attente de requêtes clientes
- La simulation peut être connectée à plusieurs clients
- Les interactions sont des réponses à des requêtes client
- Les communications et les synchronisations sont des réponses aux requêtes client
- Pas de requêtes $\Rightarrow$ Pas de sur-coût

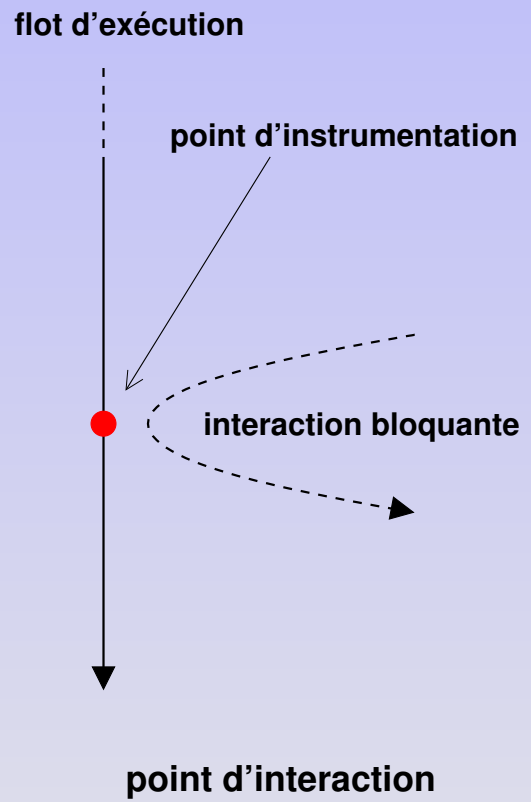
Points et plages d'interaction

Point d'interaction

- Contrôle du flux d'exécution
- Support d'interactions bloquantes
- Correspond à un point d'instrumentation: début/fin de tache, tache ponctuelle

Plage d'interaction

- Support d'interactions concurrentes à la simulation
- Délimitée par 2 points d'instrumentation
- Nous ne considérons que les plages correspondant aux tâches
- ⇒ Permet du recouvrement



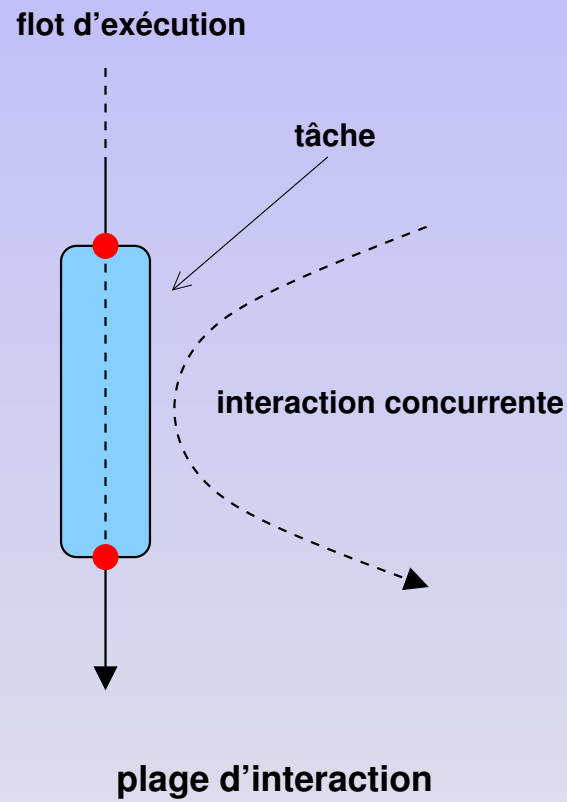
Points et plages d'interaction

Point d'interaction

- Contrôle du flux d'exécution
- Support d'interactions bloquantes
- Correspond à un point d'instrumentation: début/fin de tache, tache ponctuelle

Plage d'interaction

- Support d'interactions concurrentes à la simulation
- Délimitée par 2 points d'instrumentation
- Nous ne considérons que les plages correspondant aux tâches
- $\Rightarrow$ Permet du recouvrement



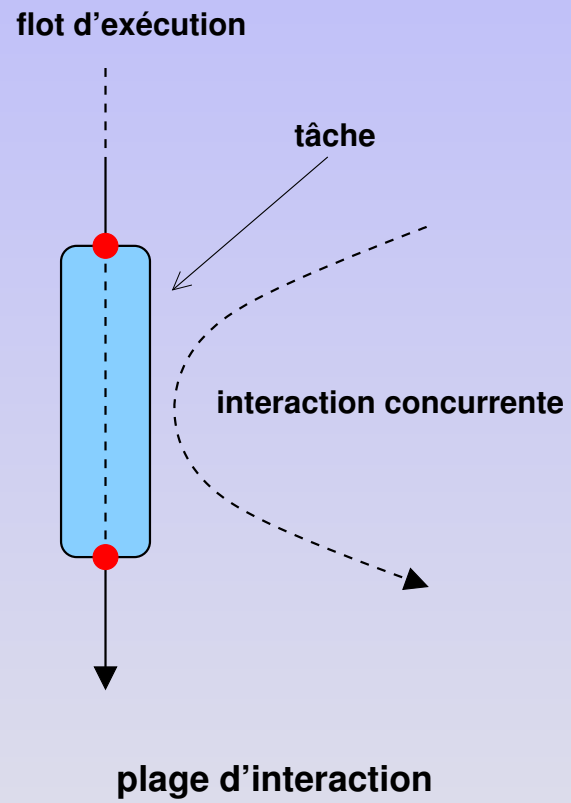
Points et plages d'interaction

Point d'interaction

- Contrôle du flux d'exécution
- Support d'interactions bloquantes
- Correspond à un point d'instrumentation: début/fin de tache, tache ponctuelle

Plage d'interaction

- Support d'interactions concurrentes à la simulation
- Délimitée par 2 points d'instrumentation
- Nous ne considérons que les plages correspondant aux tâches
- $\Rightarrow$ Permet du recouvrement



Cohérence des traitements parallèles

Organisation après la collecte

- Collecte systématique et centralisée des informations
- Organisation des informations après la collecte
- $\longrightarrow$ Limité: seulement pour le monitoring
- $\longrightarrow$ Coûteux: beaucoup de communications, centralisé

Gestion centralisée des traitements

- Synchronisations systématiques avec un serveur central
- Le serveur bloque tous les processus pour traiter les requêtes
- $\longrightarrow$ Robuste
- $\longrightarrow$ Synchronisant

Cohérence des traitements parallèles

Organisation après la collecte

- Collecte systématique et centralisée des informations
- Organisation des informations après la collecte
- $\longrightarrow$ Limité: seulement pour le monitoring
- $\longrightarrow$ Coûteux: beaucoup de communications, centralisé

Gestion centralisée des traitements

- Synchronisations systématiques avec un serveur central
- Le serveur bloque tous les processus pour traiter les requêtes
- $\longrightarrow$ Robuste
- $\longrightarrow$ Synchronisant

Cohérence des traitements parallèles

Organisation après la collecte

- Collecte systématique et centralisée des informations
- Organisation des informations après la collecte
- $\longrightarrow$ Limité: seulement pour le monitoring
- $\longrightarrow$ Coûteux: beaucoup de communications, centralisé

Gestion centralisée des traitements

- Synchronisations systématiques avec un serveur central
- Le serveur bloque tous les processus pour traiter les requêtes
- $\longrightarrow$ Robuste
- $\longrightarrow$ Synchronisant

Cohérence des traitements parallèles

Organisation après la collecte

- Collecte systématique et centralisée des informations
- Organisation des informations après la collecte
- $\longrightarrow$ Limité: seulement pour le monitoring
- $\longrightarrow$ Coûteux: beaucoup de communications, centralisé

Gestion centralisée des traitements

- Synchronisations systématiques avec un serveur central
- Le serveur bloque tous les processus pour traiter les requêtes
- $\longrightarrow$ Robuste
- $\longrightarrow$ Synchronisant

Cohérence des traitements parallèles

Organisation après la collecte

- Collecte systématique et centralisée des informations
- Organisation des informations après la collecte
- $\longrightarrow$ Limité: seulement pour le monitoring
- $\longrightarrow$ Coûteux: beaucoup de communications, centralisé

Gestion centralisée des traitements

- Synchronisations systématiques avec un serveur central
- Le serveur bloque tous les processus pour traiter les requêtes
- $\longrightarrow$ Robuste
- $\longrightarrow$ Synchronisant

Cohérence des traitements parallèles

Organisation après la collecte

- Collecte systématique et centralisée des informations
- Organisation des informations après la collecte
- $\longrightarrow$ Limité: seulement pour le monitoring
- $\longrightarrow$ Coûteux: beaucoup de communications, centralisé

Gestion centralisée des traitements

- Synchronisations systématiques avec un serveur central
- Le serveur bloque tous les processus pour traiter les requêtes
- $\longrightarrow$ Robuste
- $\longrightarrow$ Synchronisant

Cohérence des traitements parallèles

Planification des traitements

- Chaque traitement est planifié pour être exécuté à un moment précis de la simulation
- Exécution indépendante des traitements sur chaque processus
- $\longrightarrow$ Peu coûteux: pas de communications entre processus
- $\longrightarrow$ Non synchronisant

Problème

Comment déterminer une date pour la planification ?

Cohérence des traitements parallèles

Planification des traitements

- Chaque traitement est planifié pour être exécuté à un moment précis de la simulation
- Exécution indépendante des traitements sur chaque processus
- $\longrightarrow$ Peu coûteux: pas de communications entre processus
- $\longrightarrow$ Non synchronisant

Problème

Comment déterminer une date pour la planification ?

Cohérence des traitements parallèles

Planification des traitements

- Chaque traitement est planifié pour être exécuté à un moment précis de la simulation
- Exécution indépendante des traitements sur chaque processus
- $\longrightarrow$ Peu coûteux: pas de communications entre processus
- $\longrightarrow$ Non synchronisant

Problème

Comment déterminer une date pour la planification ?

Cohérence des traitements parallèles

Planification des traitements

- Chaque traitement est planifié pour être exécuté à un moment précis de la simulation
- Exécution indépendante des traitements sur chaque processus
- $\longrightarrow$ Peu coûteux: pas de communications entre processus
- $\longrightarrow$ Non synchronisant

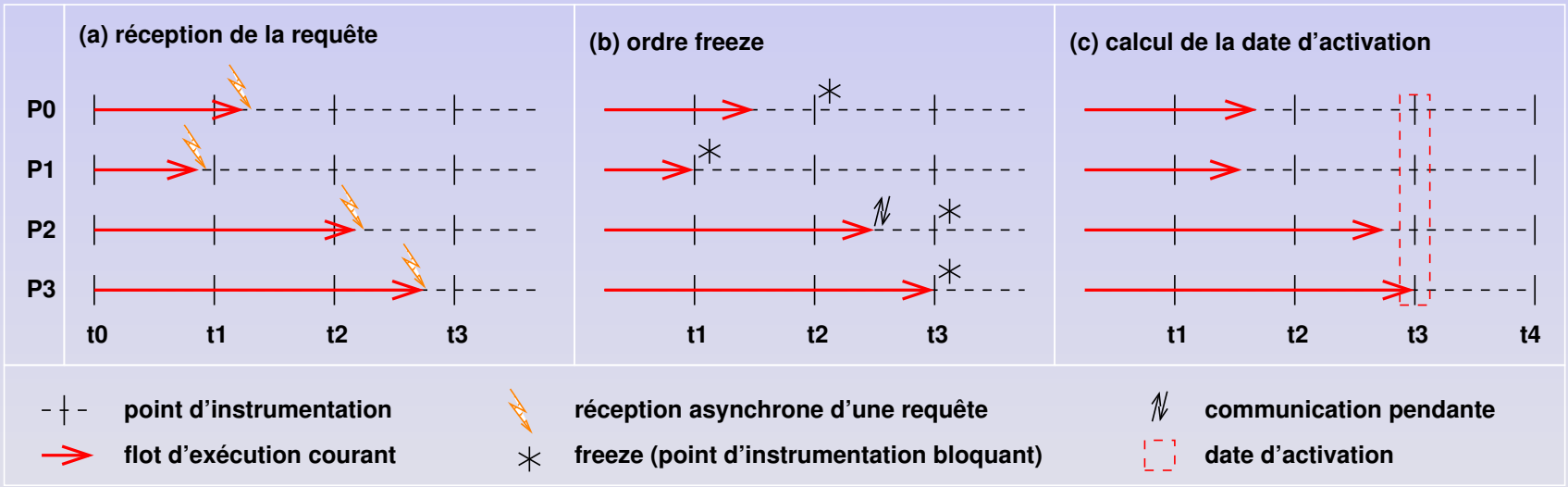
Problème

Comment déterminer une date pour la planification ?

Planification à la volée

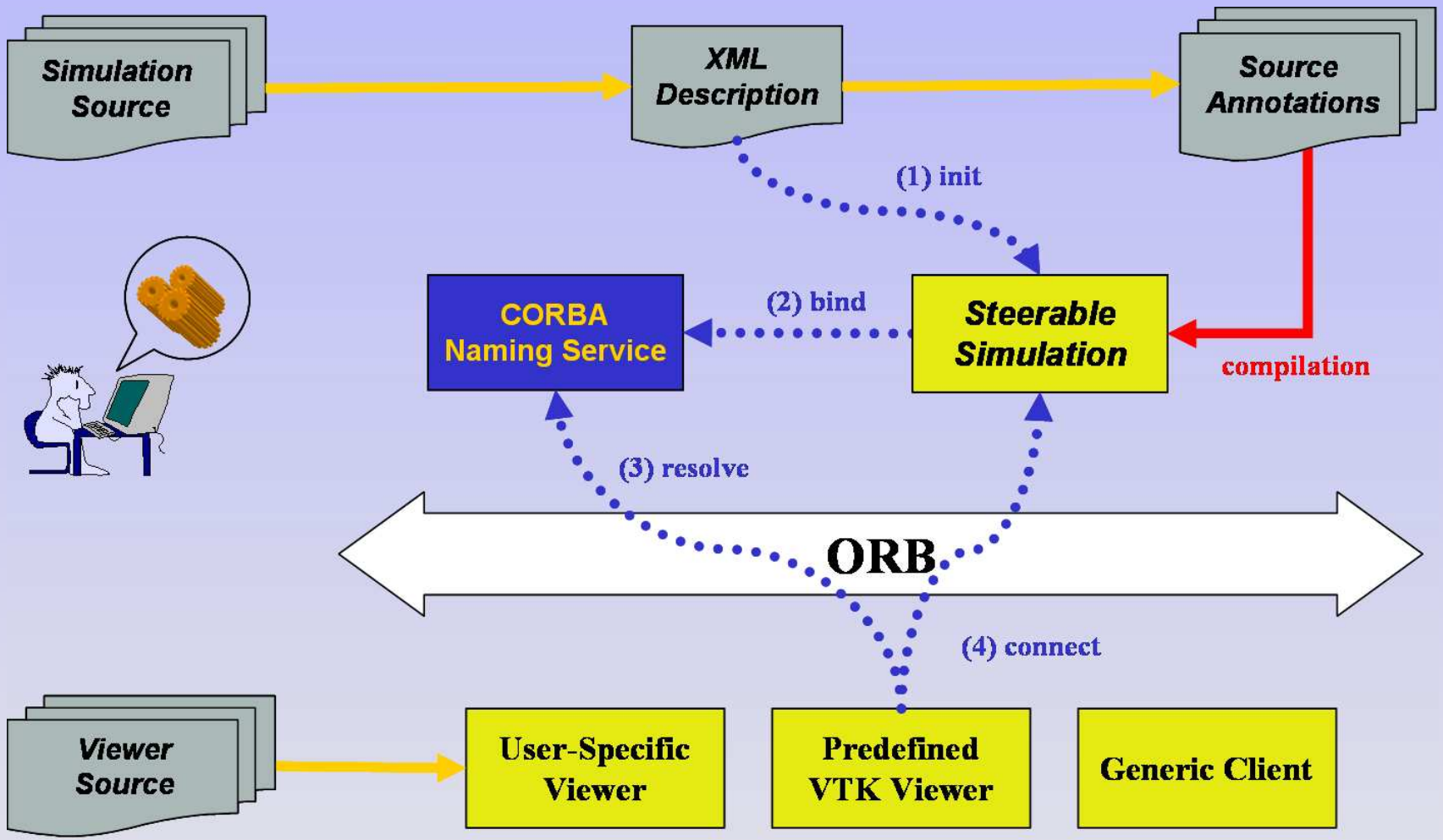
Principe

- Déterminer une date commune qu'aucun processus n'a encore dépassé
- Ajouter un critère de validité pour le début du traitement
- Comme la simulation est SPMD on obtient une date de planification valide

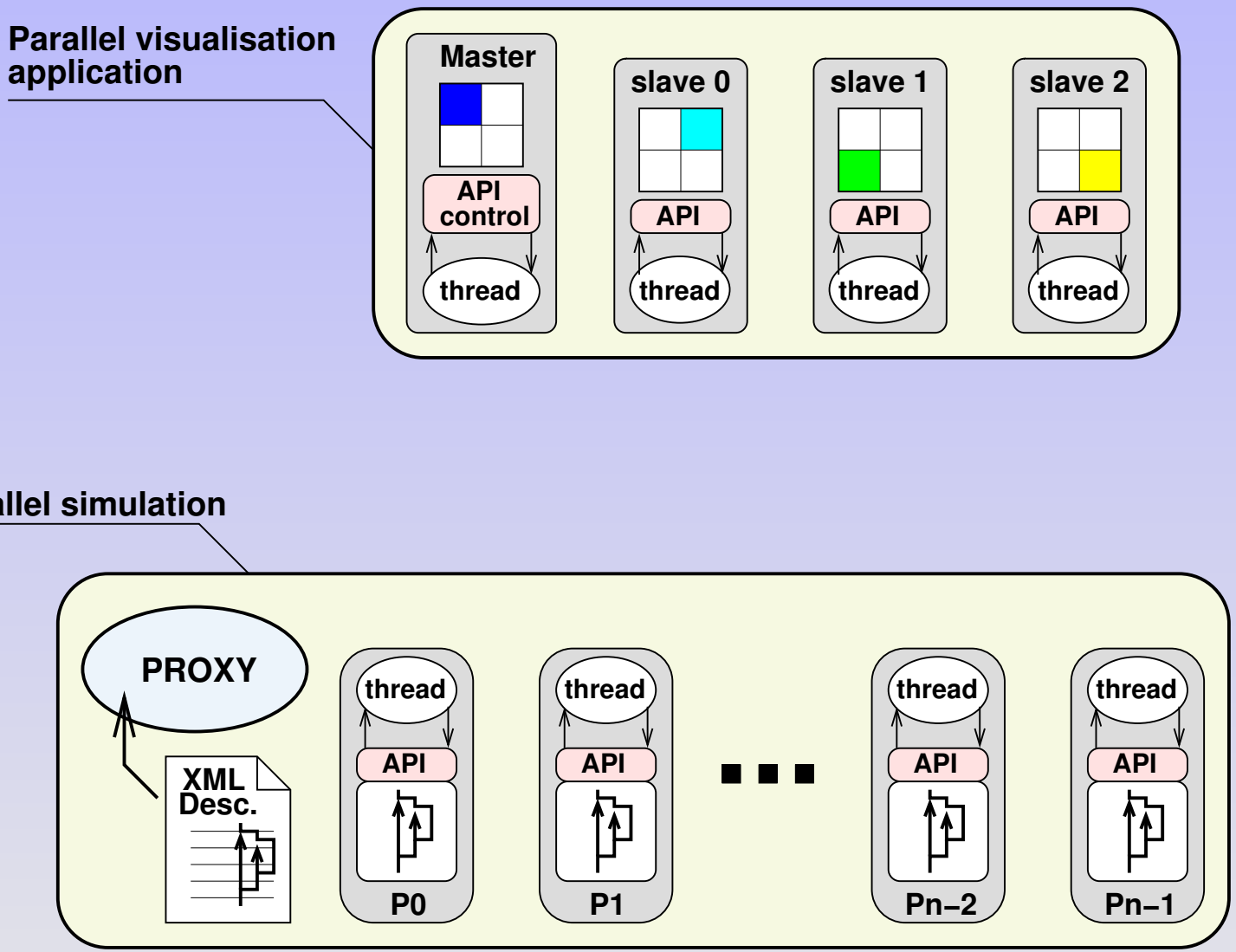


Plan

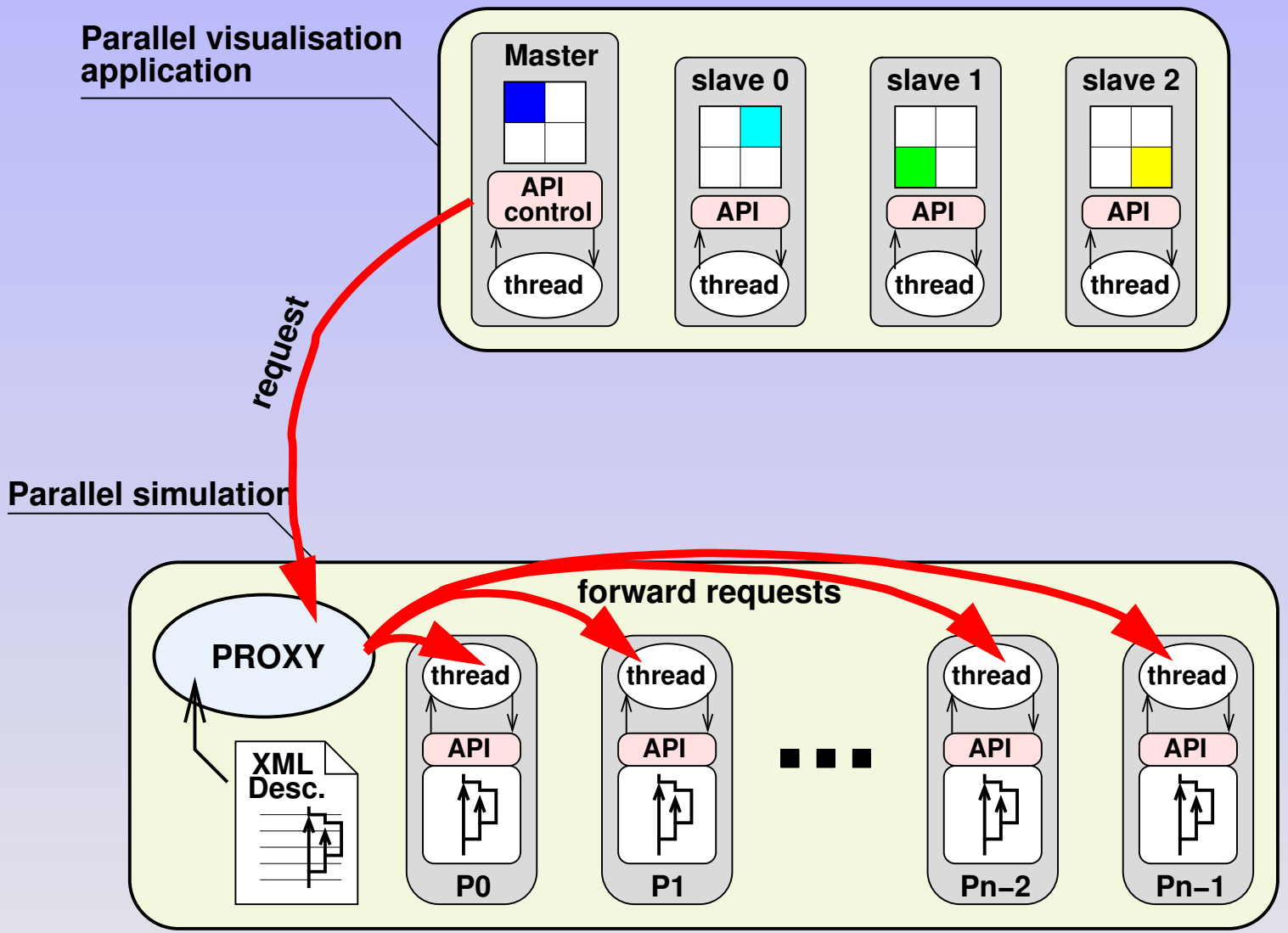
L'intégration



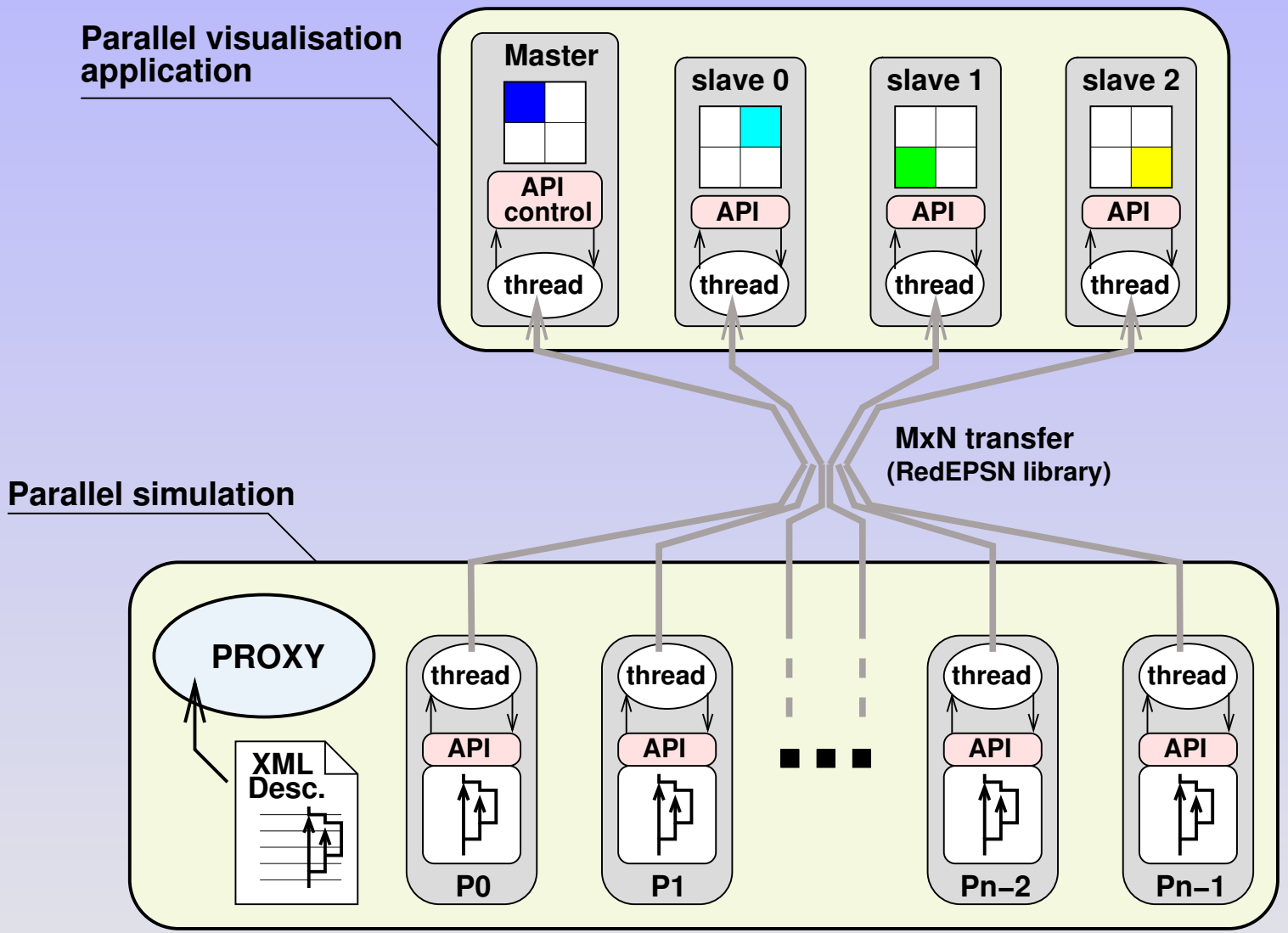
Architecture distribuée



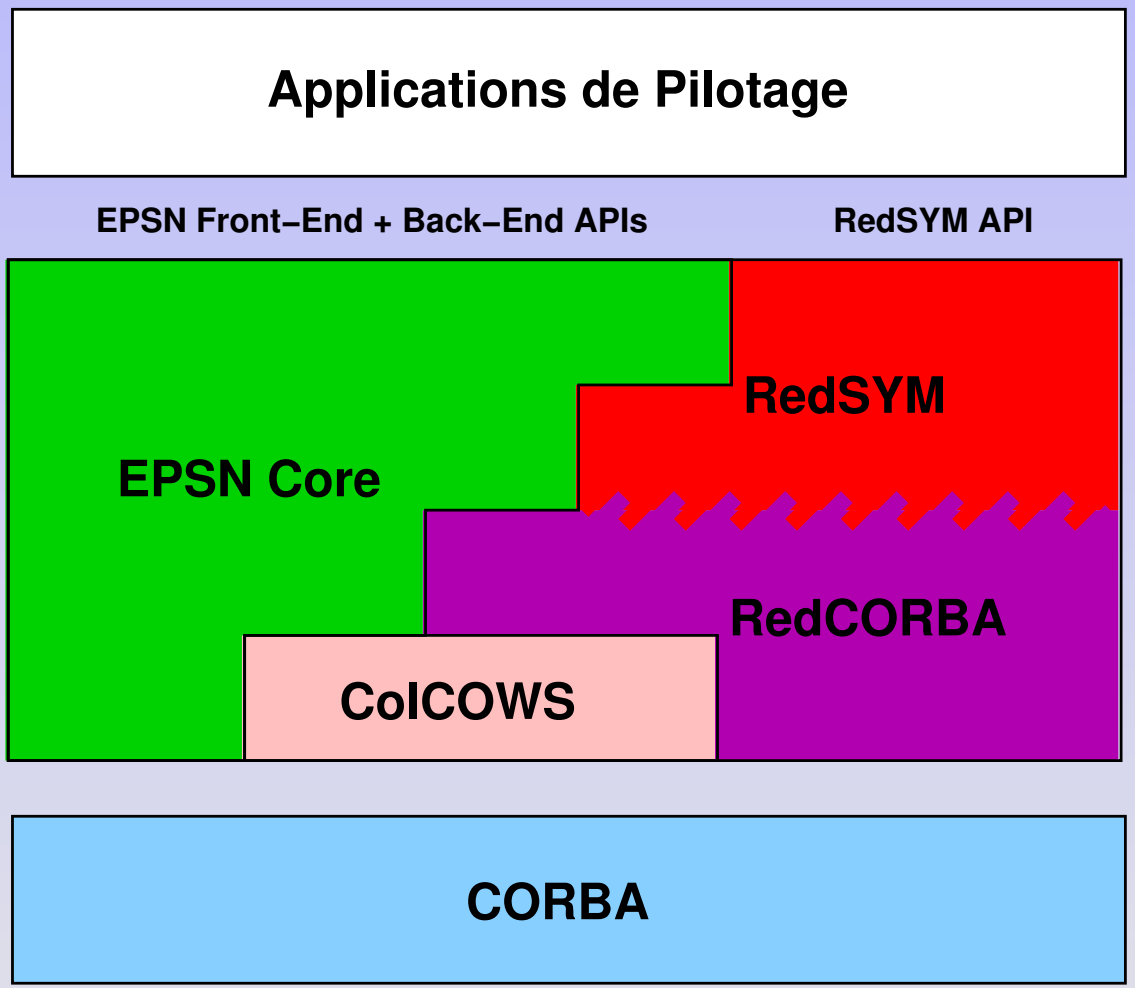
Architecture distribuée



Architecture distribuée



Les bibliothèques associées



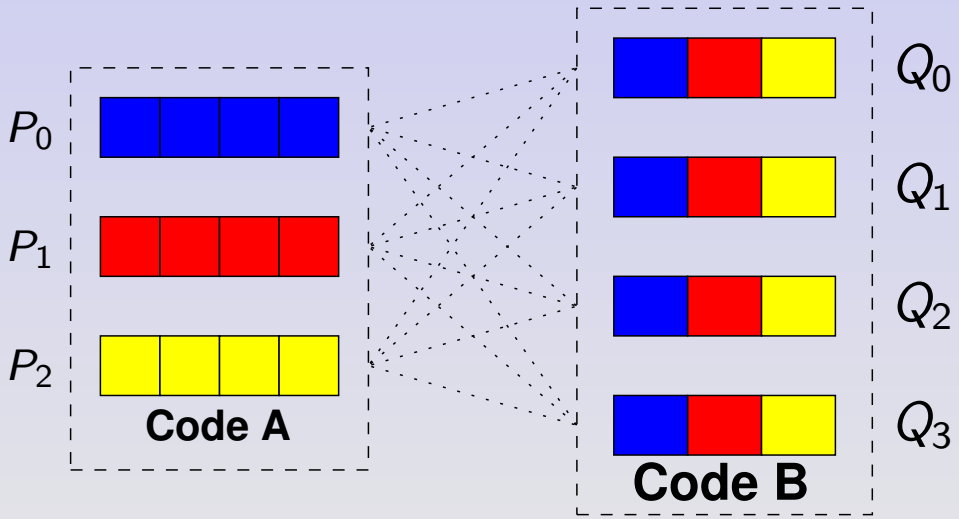
RedSYM/RedCORBA

Bibliothèque RedSYM

- Description des distribution de données
- Accès direct en mémoire
- Calcul des matrices de redistribution

Bibliothèque RedCORBA

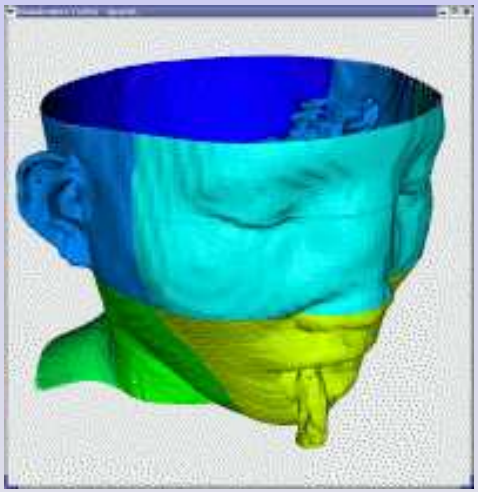
- Transfert des blocs de redistribution
- Communication via CORBA

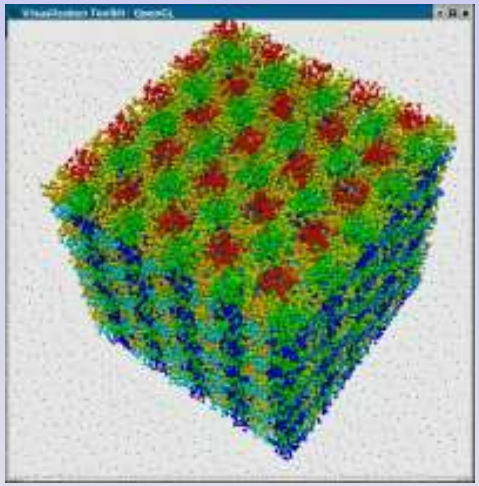


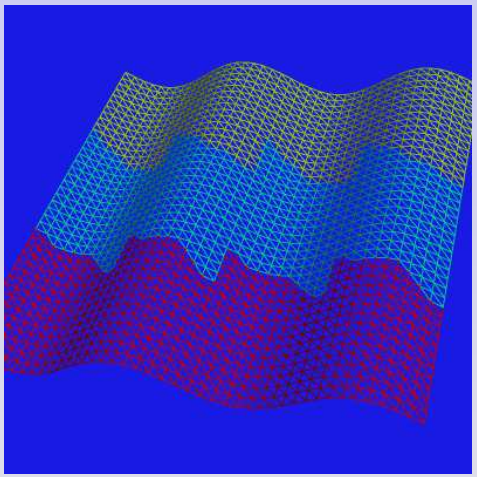
Modèle de données

Modèle de données

- Gestion de données complexes:
  scalaires, champs, points, points en blocs, maillages
- Gestion de données multi-series
- Stockage complexe (strides, structures)



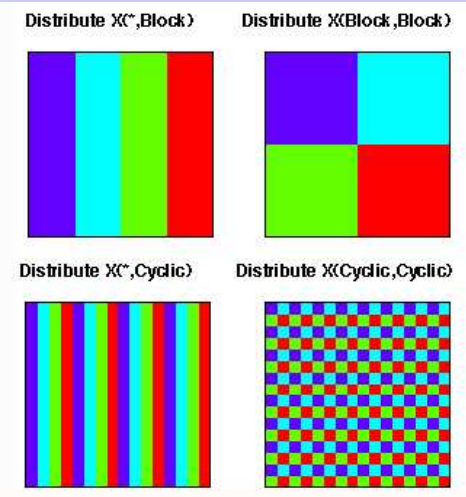


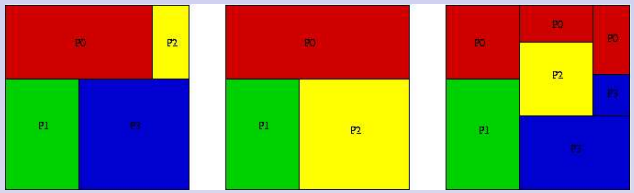


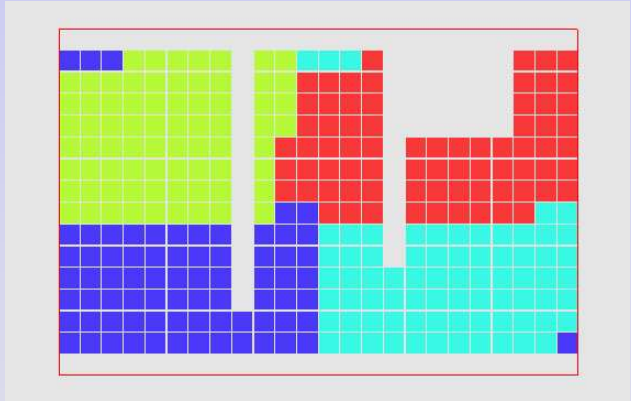
Modèle de distribution

Modèle de distribution

- Distributions bloc-cycliques à la HPF
- Distributions en blocs rectilinéaires et creux
- Partitionnement de maillage (Metis/Scotch)







Plan

Pilotage

Commandes

- Stop
  - mise en pause de l'application
  - pause sur un état cohérent
- Play
  - reprise de l'exécution
- Step
  - reprise de l'exécution pour quelques pas
  - pause sur un état cohérent

Fonctionnalités XML

- Démarrage contrôlé de l'application
- Pause automatique sur un point d'instrumentation

Accès aux données

Commandes

- Get
  - récupération ponctuelle d'une donnée de la simulation
  - récupération systématique d'une donnée avec fréquence
  - applicable à une ou toutes les séries de la donnée
- Put
  - écriture d'une donnée dans la simulation
  - applicable à une ou toutes les séries de la donnée

Fonctionnalités XML

- Gestion des droits d'accès sur les points
- Gestion des droits d'accès sur les taches

Déclenchement d'actions

Commandes

- Triger
  - déclenchement d'une action

Fonctionnalités XML

- Gestion des droits d'exécution sur les points
- Gestion des droits d'exécution sur les taches

Facilités

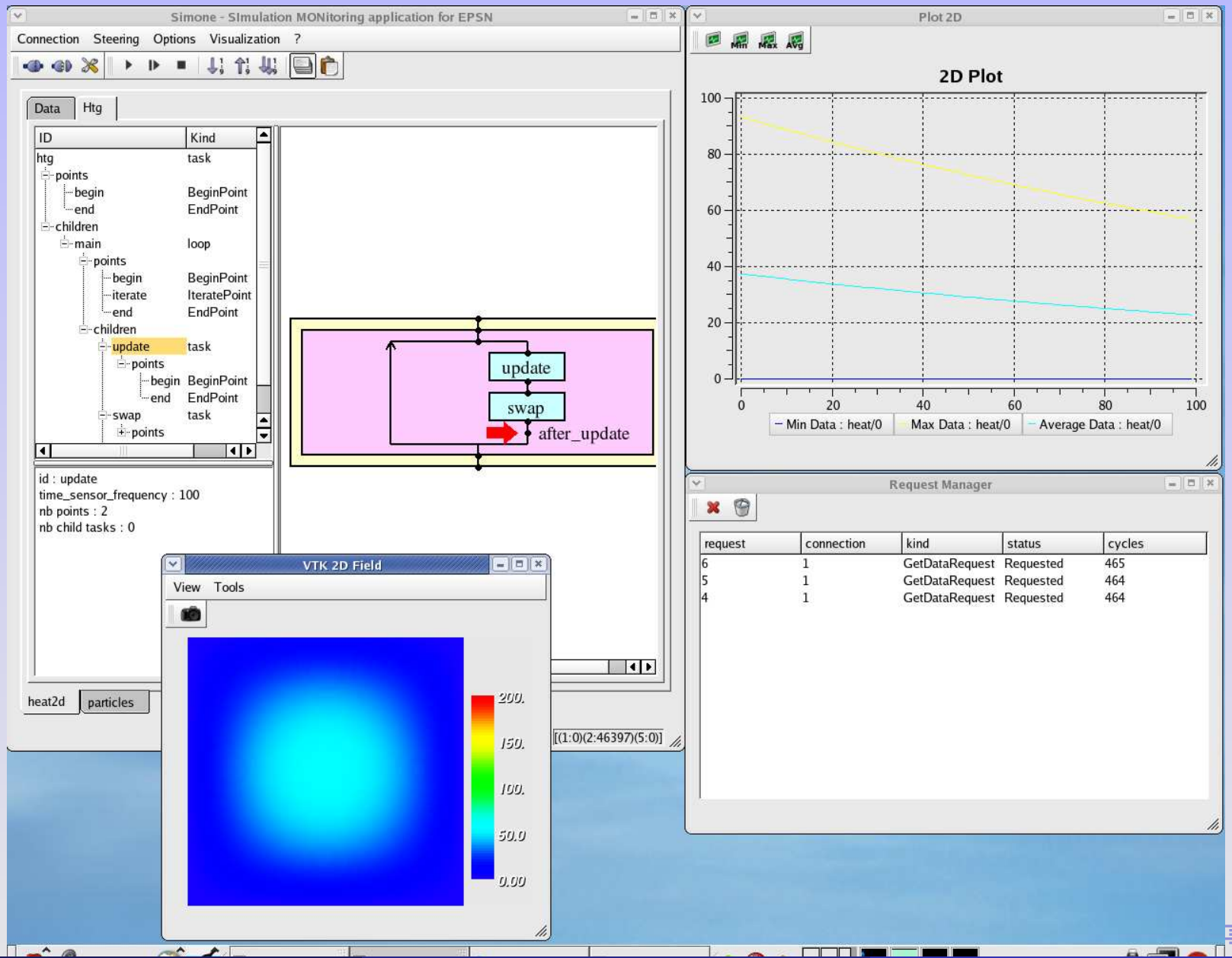
Commandes

- Benchmark
  - récupération ponctuelle des temps d'exécution d'une tache
  - récupération ponctuelle des temps d'exécution d'un point
- Date courante
  - récupération de la date courante de tous les processus
- Listes de requêtes
  - récupération de la liste de requêtes du client
  - récupération de la liste de requêtes des autres clients
  - gestion des requêtes en cours (état, suppression . . .)

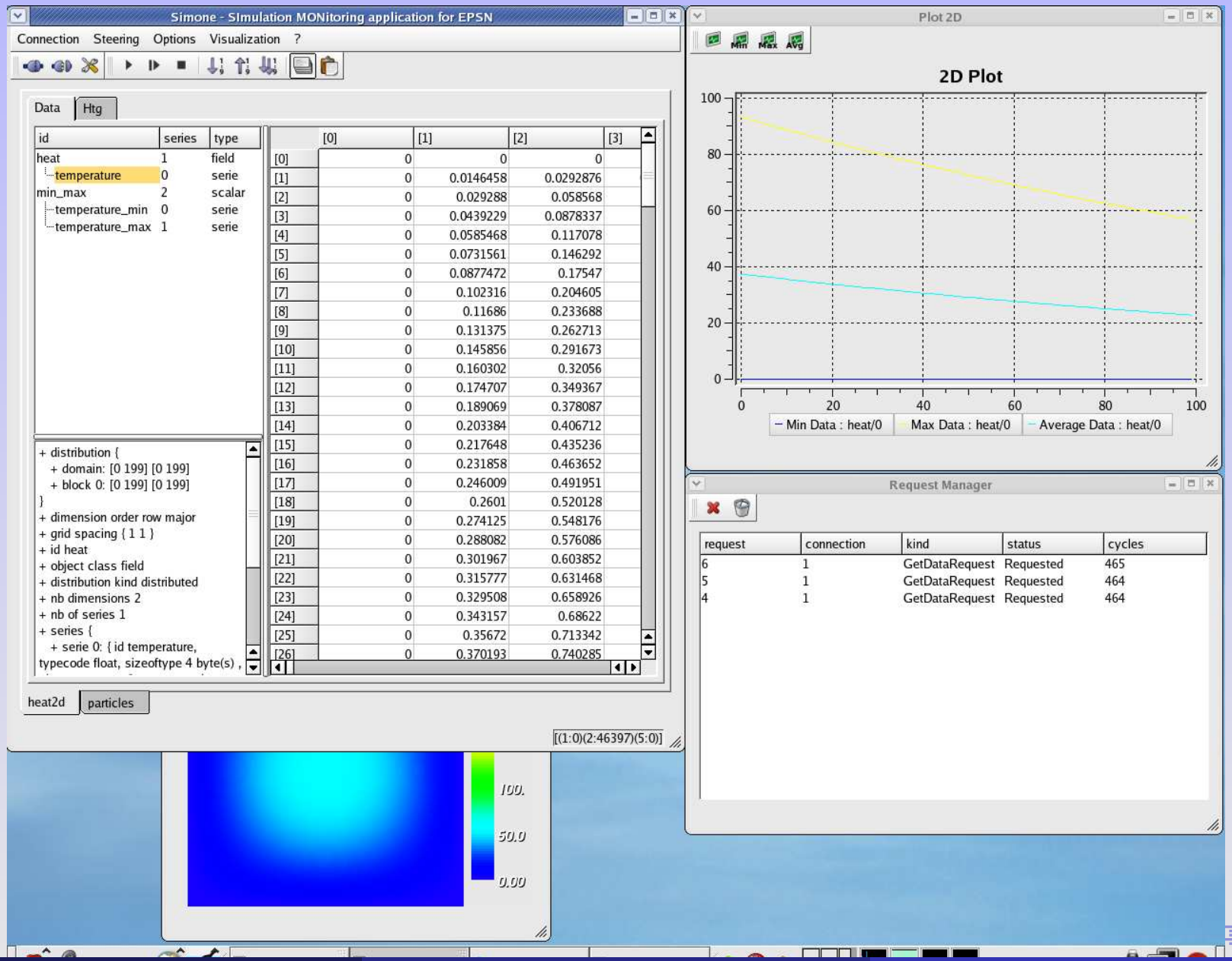
Fonctionnalités XML

- Configuration des benchmarks
  - activation par tache/point
  - réglage de la fréquence des mesures

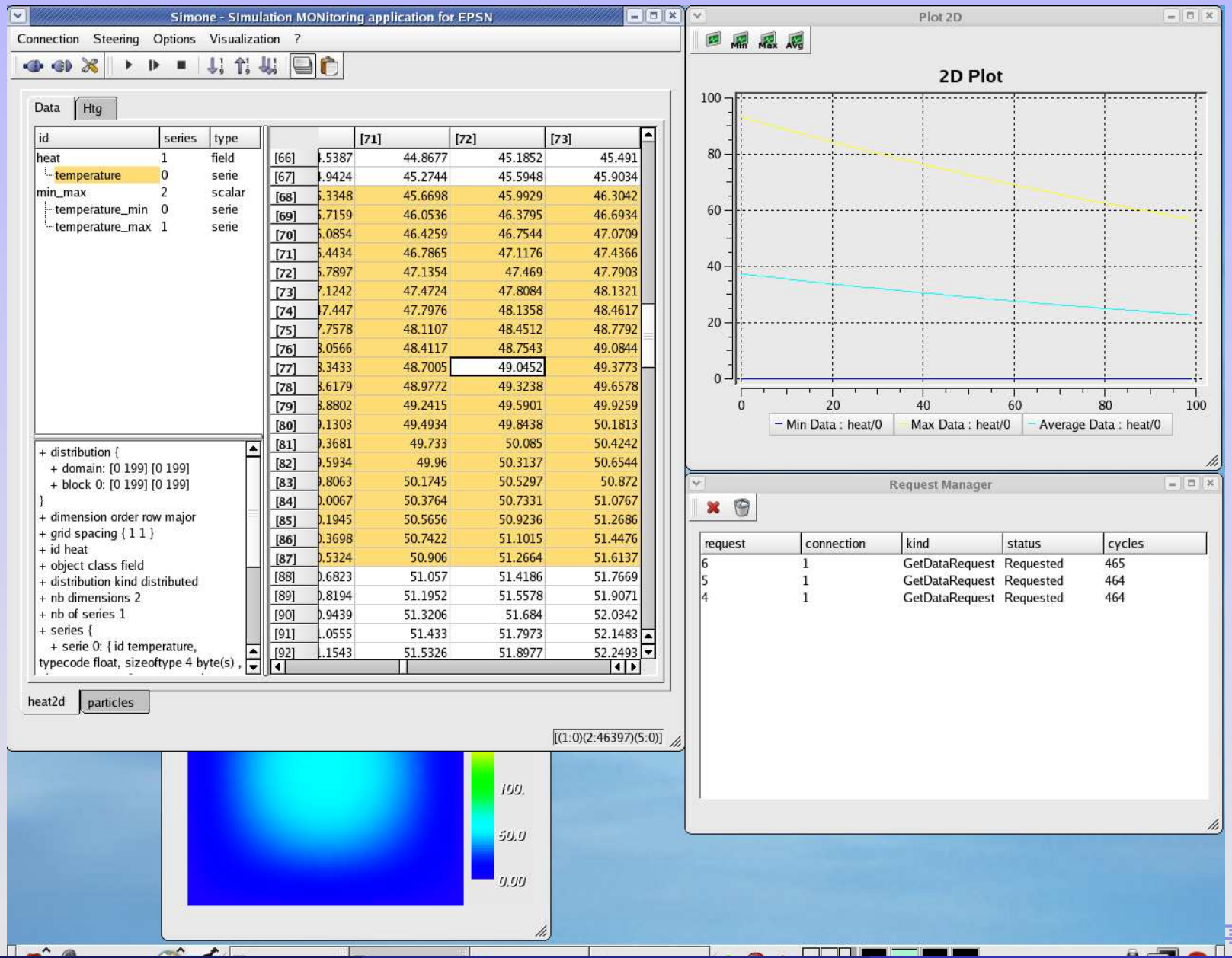
Simone


Simone - SImulation MONitoring application for EPSN
Connection Steering Options Visualization ?
Data Htg
ID Kind
htg task
points
begin BeginPoint
end EndPoint
children
main loop
points
begin BeginPoint
iterate IteratePoint
end EndPoint
children
update task
points
begin BeginPoint
end EndPoint
swap task
points
id : update
time_sensor_frequency : 100
nb points : 2
nb child tasks : 0
update
swap
after_update
heat2d particles
[(1:0)(2:46397)(5:0)]
Plot 2D
2D Plot
Min Max Avg
100
80
60
40
20
0
0 20 40 60 80 100
Min Data : heat/0
Max Data : heat/0
Average Data : heat/0
Request Manager
request connection kind status cycles
6 1 GetDataRequest Requested 465
5 1 GetDataRequest Requested 464
4 1 GetDataRequest Requested 464
VTK 2D Field
View Tools
200.
150.
100.
50.0
0.00

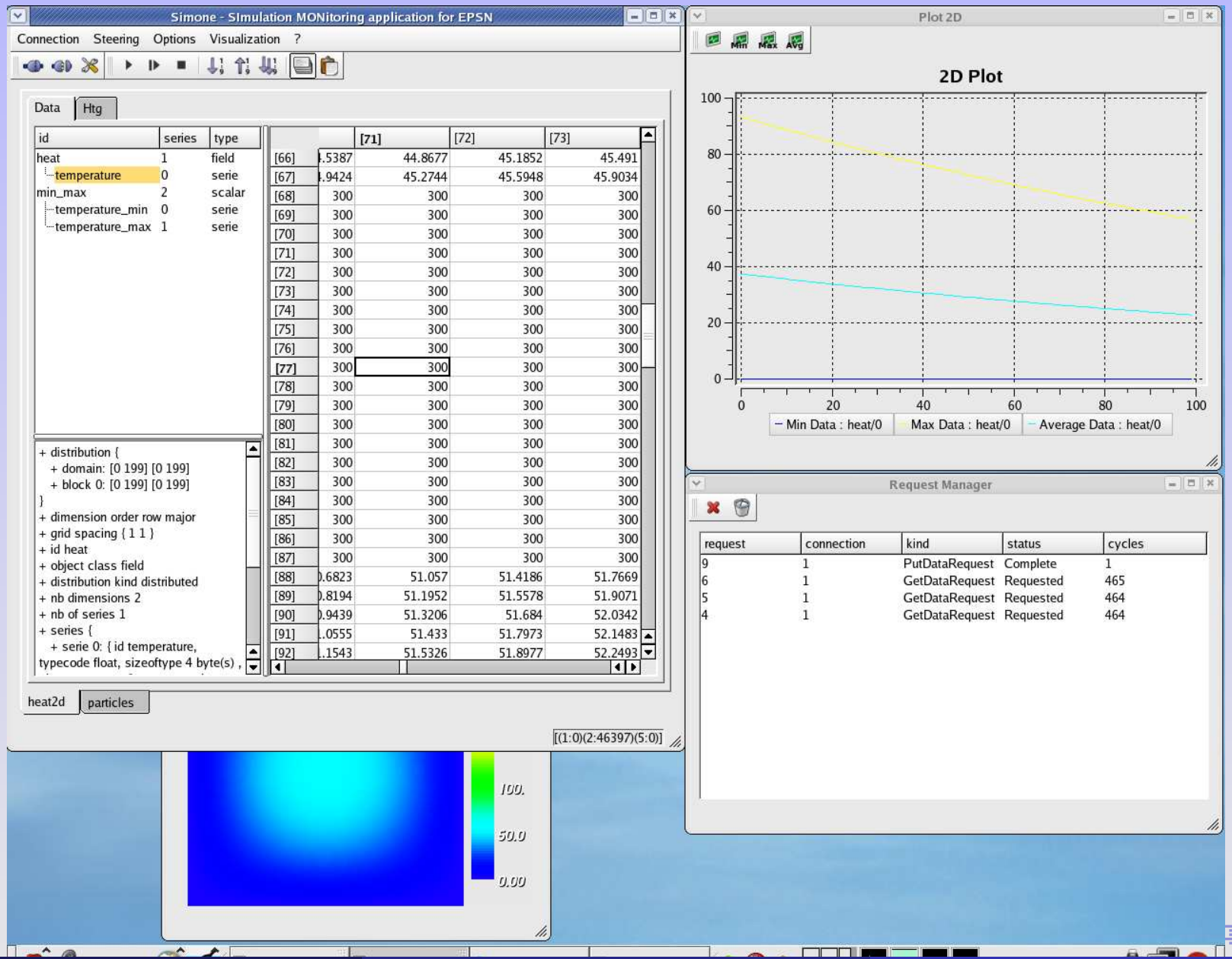
Simone


Simone - SImulation MONitoring application for EPSN
Connection Steering Options Visualization ?
Data Htg
id series type
heat 1 field
temperature 0 serie
min_max 2 scalar
temperature_min 0 serie
temperature_max 1 serie
[0] [1] [2] [3]
+ distribution {
+ domain: [0 199] [0 199]
+ block 0: [0 199] [0 199]
}
+ dimension order row major
+ grid spacing { 1 1 }
+ id heat
+ object class field
+ distribution kind distributed
+ nb dimensions 2
+ nb of series 1
+ series {
+ serie 0: { id temperature,
typecode float, sizeoftype 4 byte(s) ,
heat2d particles
[(1:0)(2:46397)(5:0)]
Plot 2D
2D Plot
Min Data : heat/0
Max Data : heat/0
Average Data : heat/0
Request Manager
request connection kind status cycles
6 1 GetDataRequest Requested 465
5 1 GetDataRequest Requested 464
4 1 GetDataRequest Requested 464
100.
50.0
0.00

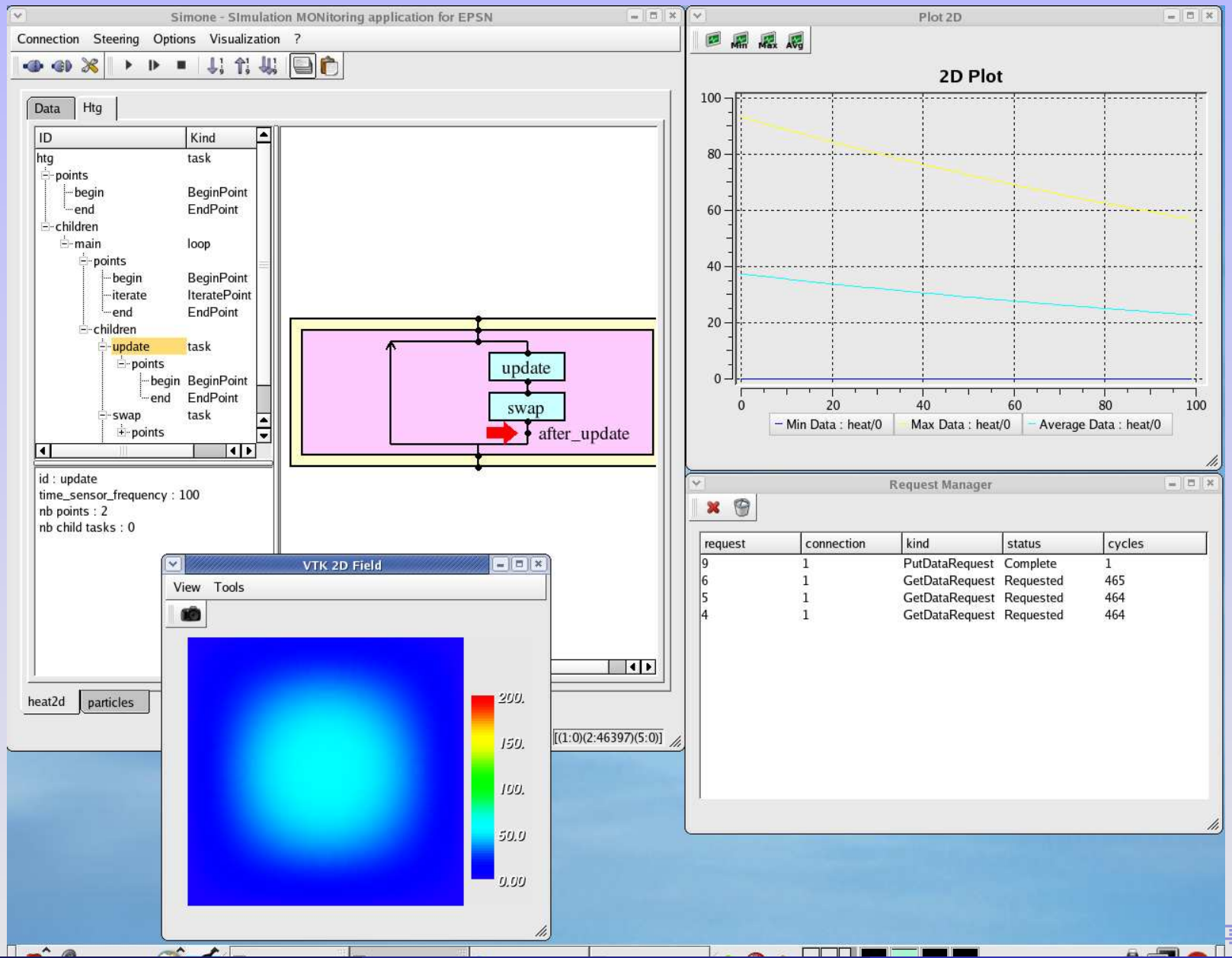
Simone


Simone - SImulation MONitoring application for EPSN
Connection Steering Options Visualization ?
Data Htg
id series type
heat 1 field
temperature 0 serie
min_max 2 scalar
temperature_min 0 serie
temperature_max 1 serie
[71] [72] [73]
+ distribution {
+ domain: [0 199] [0 199]
+ block 0: [0 199] [0 199]
}
+ dimension order row major
+ grid spacing { 1 1 }
+ id heat
+ object class field
+ distribution kind distributed
+ nb dimensions 2
+ nb of series 1
+ series {
+ serie 0: { id temperature,
typecode float, sizeoftype 4 byte(s) ,
heat2d particles
[(1:0)(2:46397)(5:0)]
Plot 2D
2D Plot
Min Data : heat/0
Max Data : heat/0
Average Data : heat/0
Request Manager
request connection kind status cycles
6 1 GetDataRequest Requested 465
5 1 GetDataRequest Requested 464
4 1 GetDataRequest Requested 464
100.
50.0
0.00

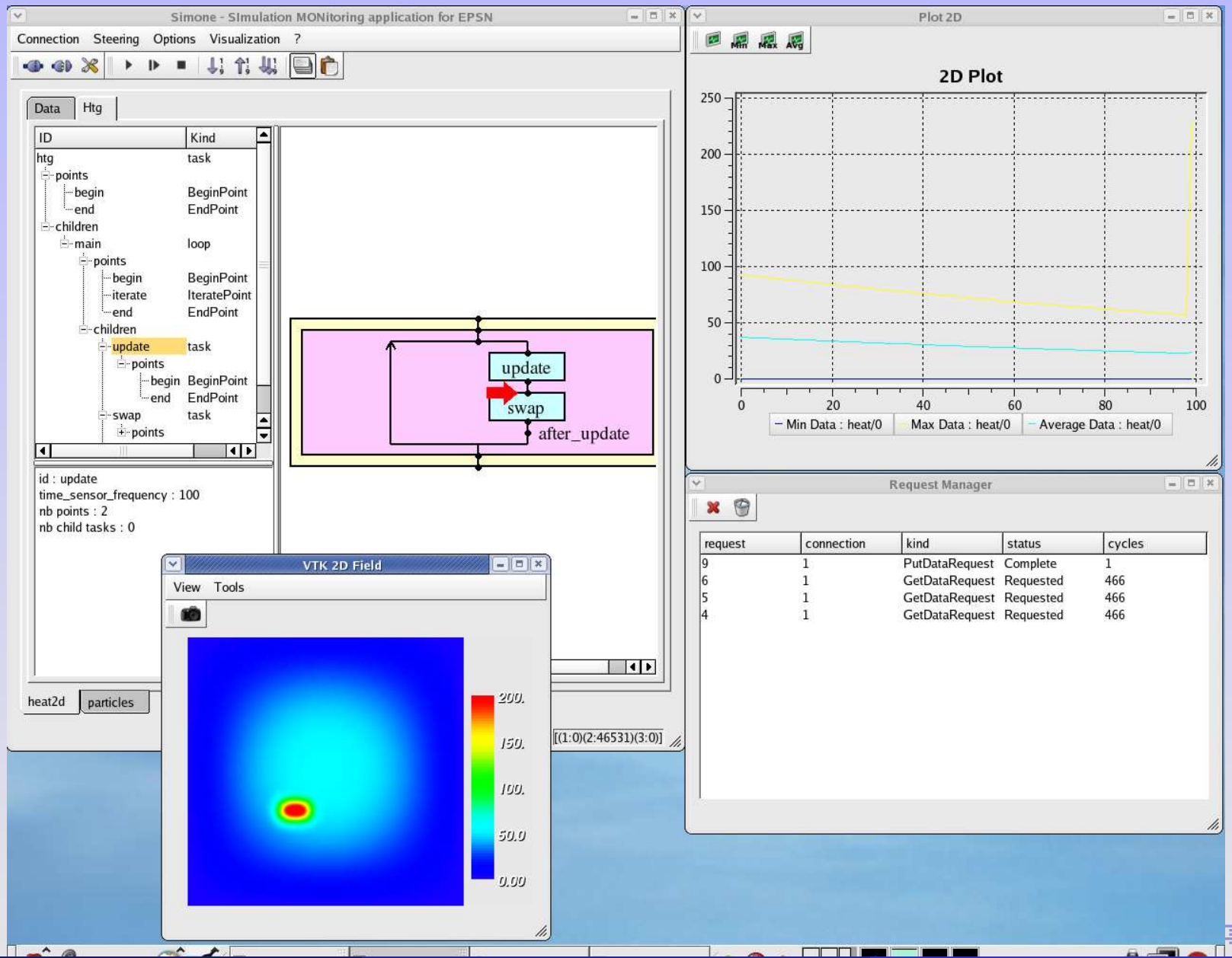
Simone


Simone - SImulation MONitoring application for EPSN
Connection Steering Options Visualization ?
Data Htg
id series type
heat 1 field
temperature 0 serie
min_max 2 scalar
temperature_min 0 serie
temperature_max 1 serie
+ distribution {
+ domain: [0 199] [0 199]
+ block 0: [0 199] [0 199]
}
+ dimension order row major
+ grid spacing { 1 1 }
+ id heat
+ object class field
+ distribution kind distributed
+ nb dimensions 2
+ nb of series 1
+ series {
+ serie 0: { id temperature,
typecode float, sizeoftype 4 byte(s) ,
heat2d particles
[(1:0)(2:46397)(5:0)]
Plot 2D
2D Plot
Min Data : heat/0
Max Data : heat/0
Average Data : heat/0
Request Manager
request connection kind status cycles
9 1 PutDataRequest Complete 1
6 1 GetDataRequest Requested 465
5 1 GetDataRequest Requested 464
4 1 GetDataRequest Requested 464
100.
50.0
0.00

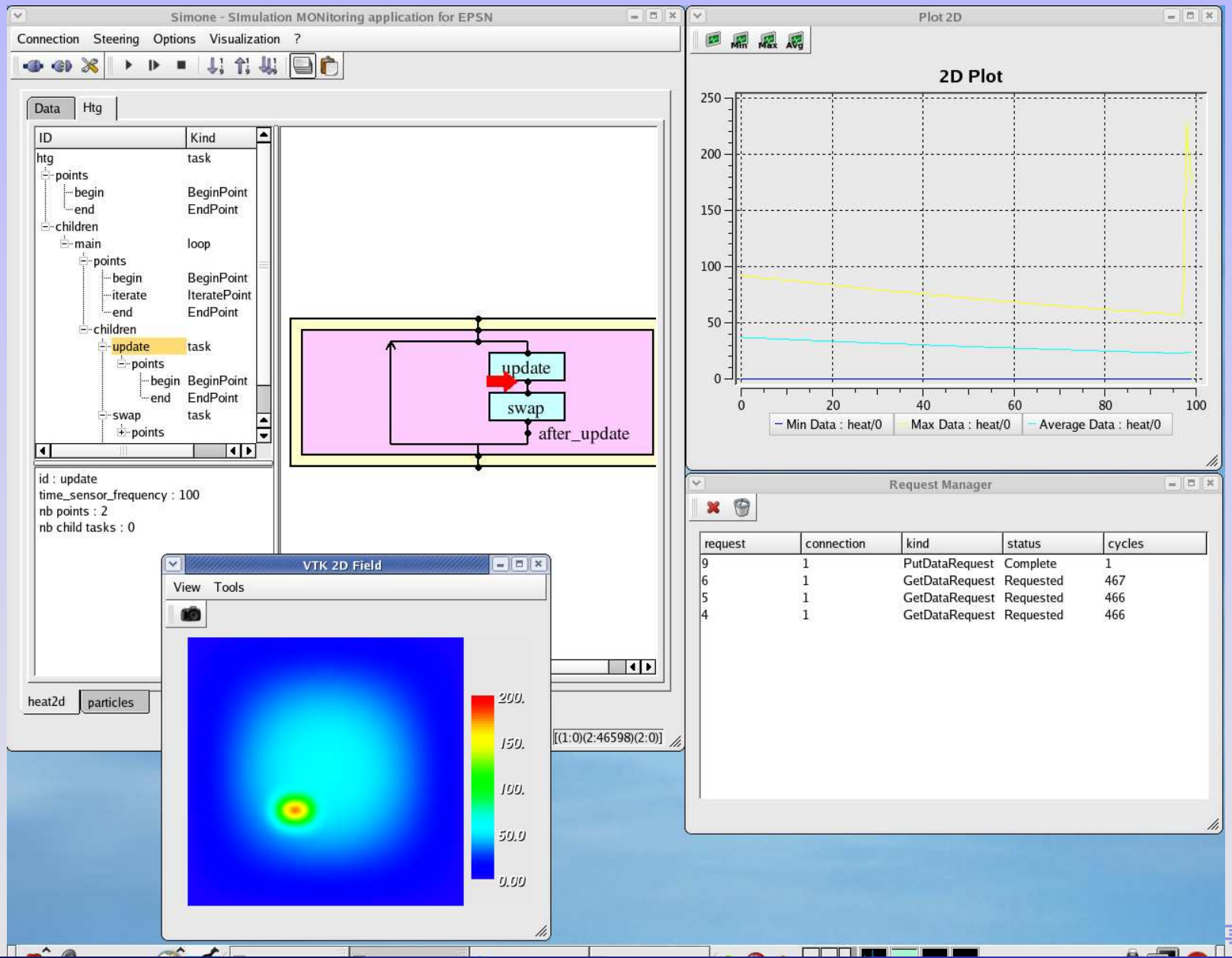
Simone


Simone - SImulation MONitoring application for EPSN
Connection Steering Options Visualization ?
Data Htg
ID Kind
htg task
points
begin BeginPoint
end EndPoint
children
main loop
points
begin BeginPoint
iterate IteratePoint
end EndPoint
children
update task
points
begin BeginPoint
end EndPoint
swap task
points
id : update
time_sensor_frequency : 100
nb points : 2
nb child tasks : 0
update
swap
after_update
heat2d particles
[(1:0)(2:46397)(5:0)]
Plot 2D
2D Plot
Min Max Avg
100
80
60
40
20
0
0 20 40 60 80 100
Min Data : heat/0
Max Data : heat/0
Average Data : heat/0
Request Manager
request connection kind status cycles
9 1 PutDataRequest Complete 1
6 1 GetDataRequest Requested 465
5 1 GetDataRequest Requested 464
4 1 GetDataRequest Requested 464
VTK 2D Field
View Tools
200.
150.
100.
50.0
0.00

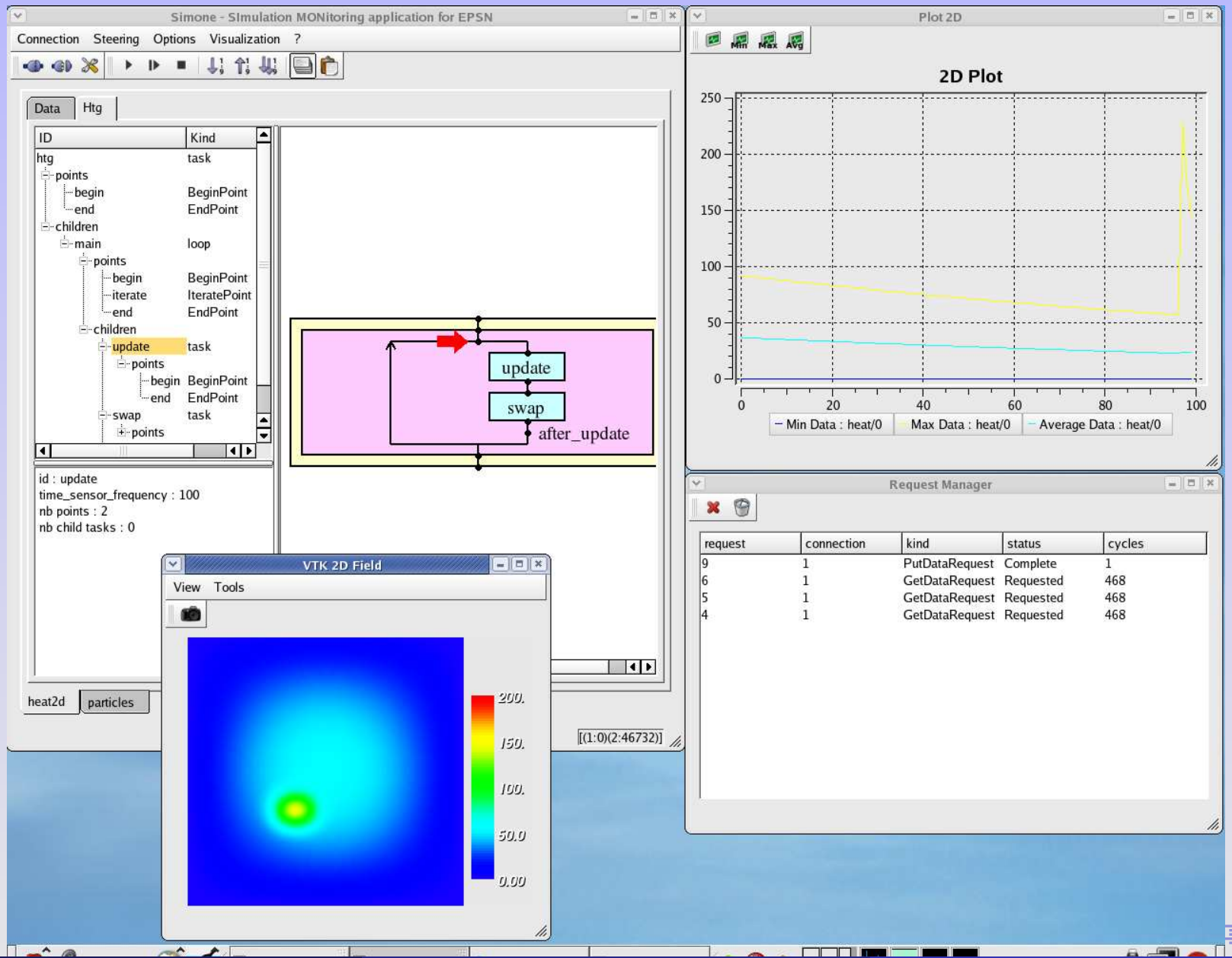
Simone


Simone - SImulation MONitoring application for EPSN
Connection Steering Options Visualization ?
Data Htg
ID Kind
htg task
points
begin BeginPoint
end EndPoint
children
main loop
points
begin BeginPoint
iterate IteratePoint
end EndPoint
children
update task
points
begin BeginPoint
end EndPoint
swap task
points
id : update
time_sensor_frequency : 100
nb points : 2
nb child tasks : 0
update
swap
after_update
heat2d particles
[(1:0)(2:46531)(3:0)]
Plot 2D
2D Plot
250
200
150
100
50
0
0 20 40 60 80 100
Min Data : heat/0
Max Data : heat/0
Average Data : heat/0
Request Manager
request connection kind status cycles
9 1 PutDataRequest Complete 1
6 1 GetDataRequest Requested 466
5 1 GetDataRequest Requested 466
4 1 GetDataRequest Requested 466
VTK 2D Field
View Tools
200.
150.
100.
50.0
0.00

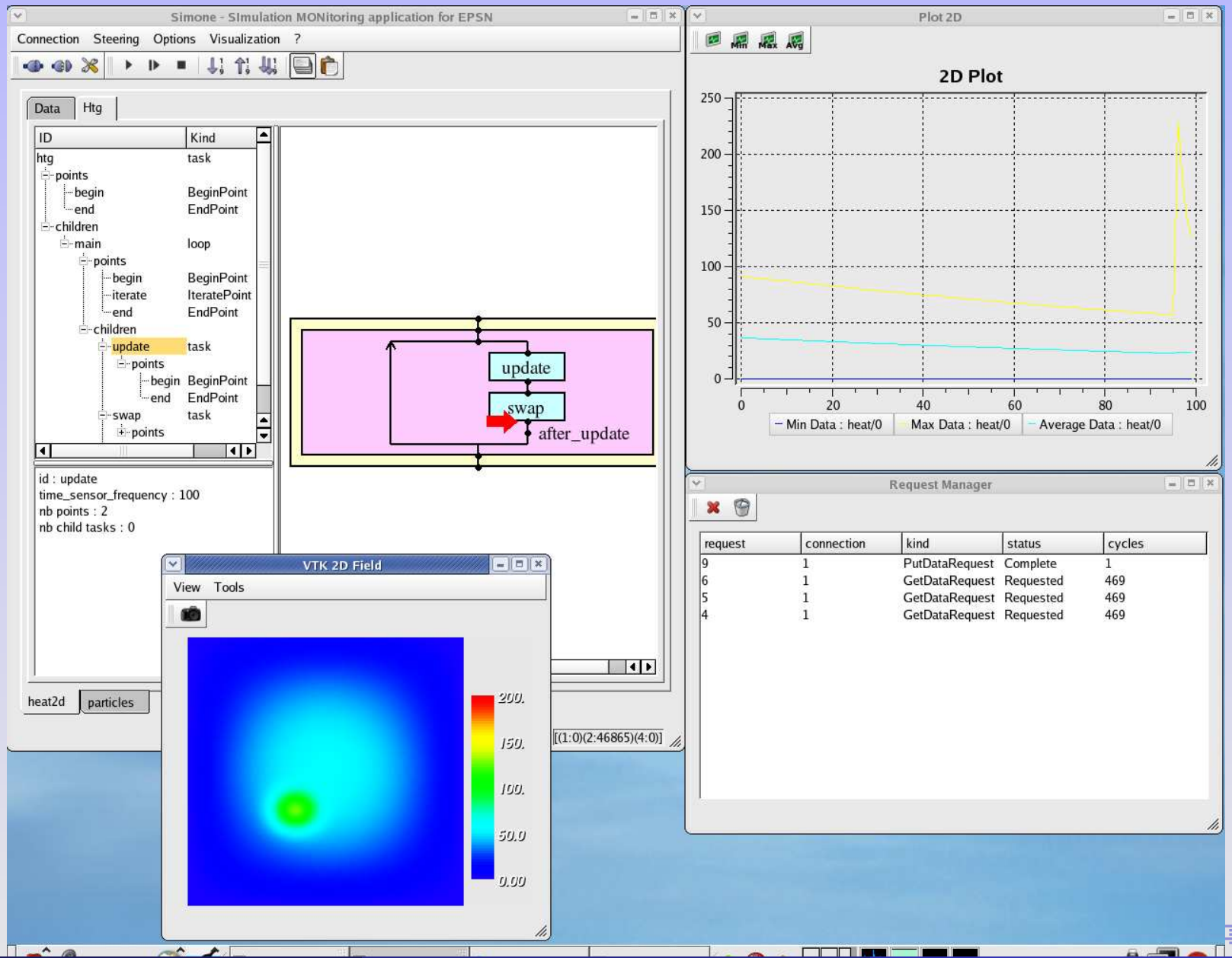
Simone


Simone - SImulation MONitoring application for EPSN
Connection Steering Options Visualization ?
Data Htg
ID Kind
htg task
points
begin BeginPoint
end EndPoint
children
main loop
points
begin BeginPoint
iterate IteratePoint
end EndPoint
children
update task
points
begin BeginPoint
end EndPoint
swap task
points
id : update
time_sensor_frequency : 100
nb points : 2
nb child tasks : 0
update
swap
after_update
heat2d particles
[(1:0)(2:46598)(2:0)]
Plot 2D
Min Max Avg
2D Plot
250 200 150 100 50 0
0 20 40 60 80 100
Min Data : heat/0
Max Data : heat/0
Average Data : heat/0
Request Manager
request connection kind status cycles
9 1 PutDataRequest Complete 1
6 1 GetDataRequest Requested 467
5 1 GetDataRequest Requested 466
4 1 GetDataRequest Requested 466
VTK 2D Field
View Tools
200.
150.
100.
50.0
0.00

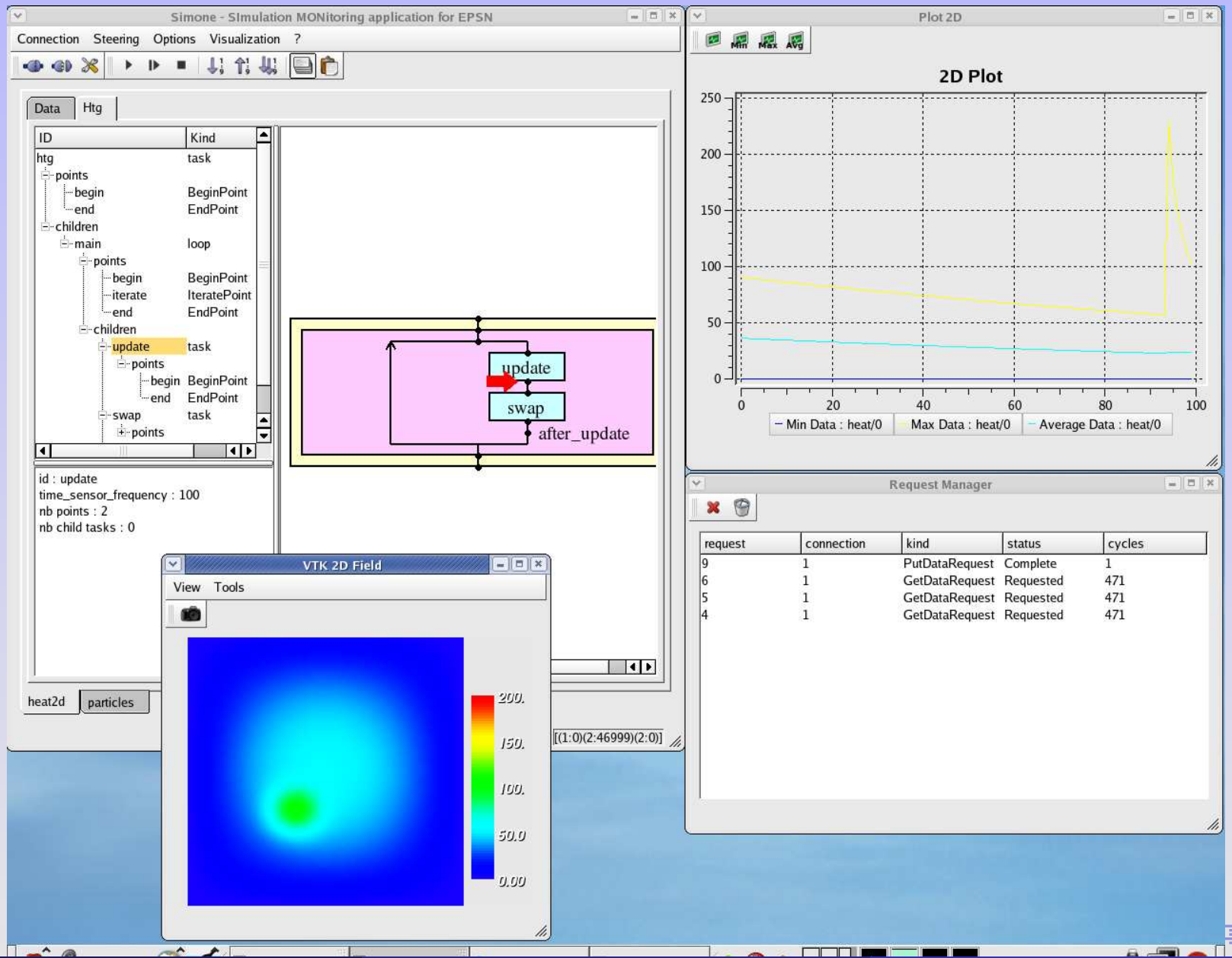
Simone


Simone - SImulation MONitoring application for EPSN
Connection Steering Options Visualization ?
Data Htg
ID Kind
htg task
points
begin BeginPoint
end EndPoint
children
main loop
points
begin BeginPoint
iterate IteratePoint
end EndPoint
children
update task
points
begin BeginPoint
end EndPoint
swap task
points
id : update
time_sensor_frequency : 100
nb points : 2
nb child tasks : 0
update
swap
after_update
heat2d particles
[(1:0)(2:46732)]
Plot 2D
2D Plot
250
200
150
100
50
0
0 20 40 60 80 100
Min Data : heat/0
Max Data : heat/0
Average Data : heat/0
Request Manager
request connection kind status cycles
9 1 PutDataRequest Complete 1
6 1 GetDataRequest Requested 468
5 1 GetDataRequest Requested 468
4 1 GetDataRequest Requested 468
VTK 2D Field
View Tools
200.
150.
100.
50.0
0.00

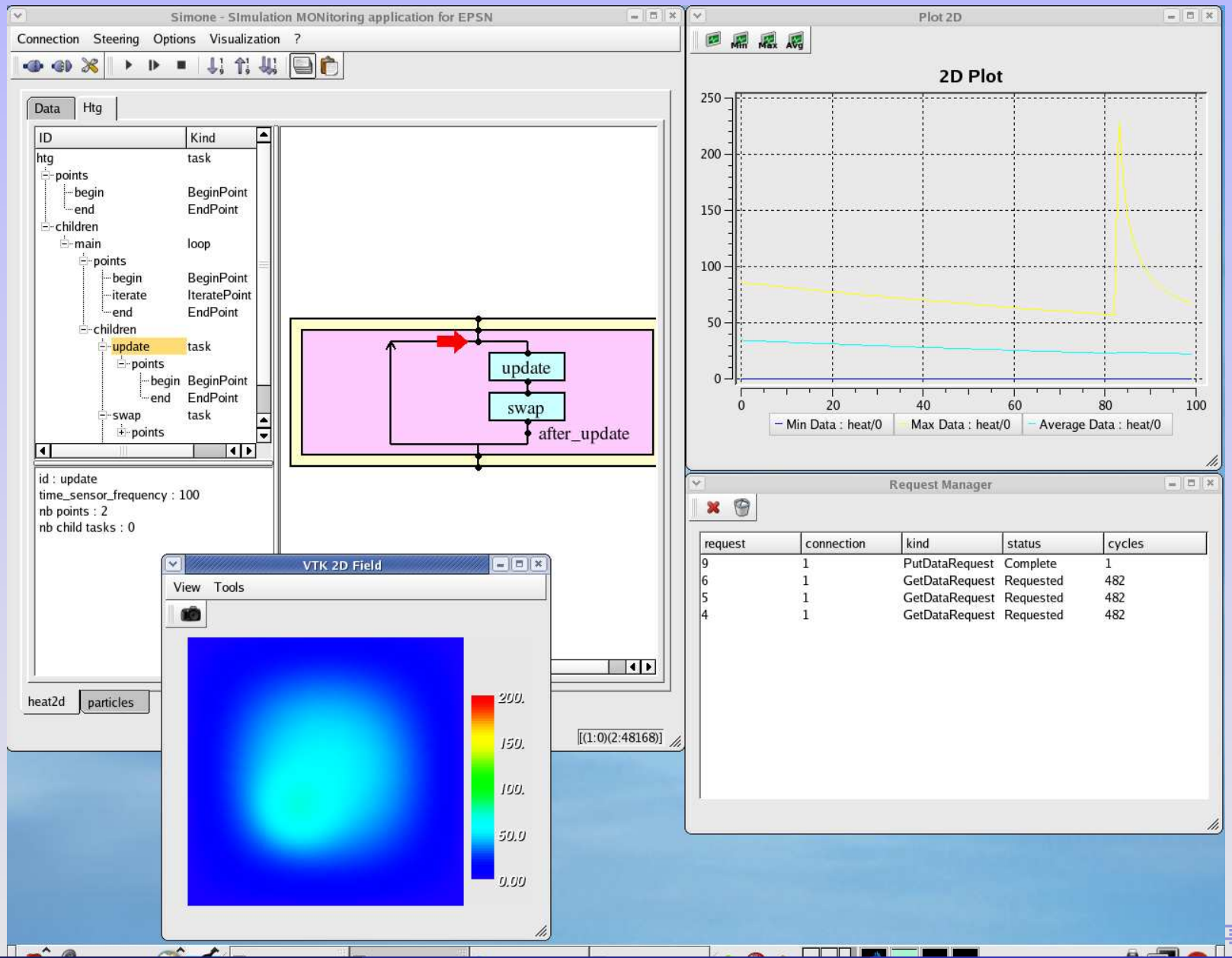
Simone


Simone - SImulation MONitoring application for EPSN
Connection Steering Options Visualization ?
Data Htg
ID Kind
htg task
points
begin BeginPoint
end EndPoint
children
main loop
points
begin BeginPoint
iterate IteratePoint
end EndPoint
children
update task
points
begin BeginPoint
end EndPoint
swap task
points
id : update
time_sensor_frequency : 100
nb points : 2
nb child tasks : 0
update
swap
after_update
heat2d particles
[(1:0)(2:46865)(4:0)]
Plot 2D
2D Plot
250
200
150
100
50
0
0 20 40 60 80 100
Min Data : heat/0
Max Data : heat/0
Average Data : heat/0
Request Manager
request connection kind status cycles
9 1 PutDataRequest Complete 1
6 1 GetDataRequest Requested 469
5 1 GetDataRequest Requested 469
4 1 GetDataRequest Requested 469
VTK 2D Field
View Tools
200.
150.
100.
50.0
0.00

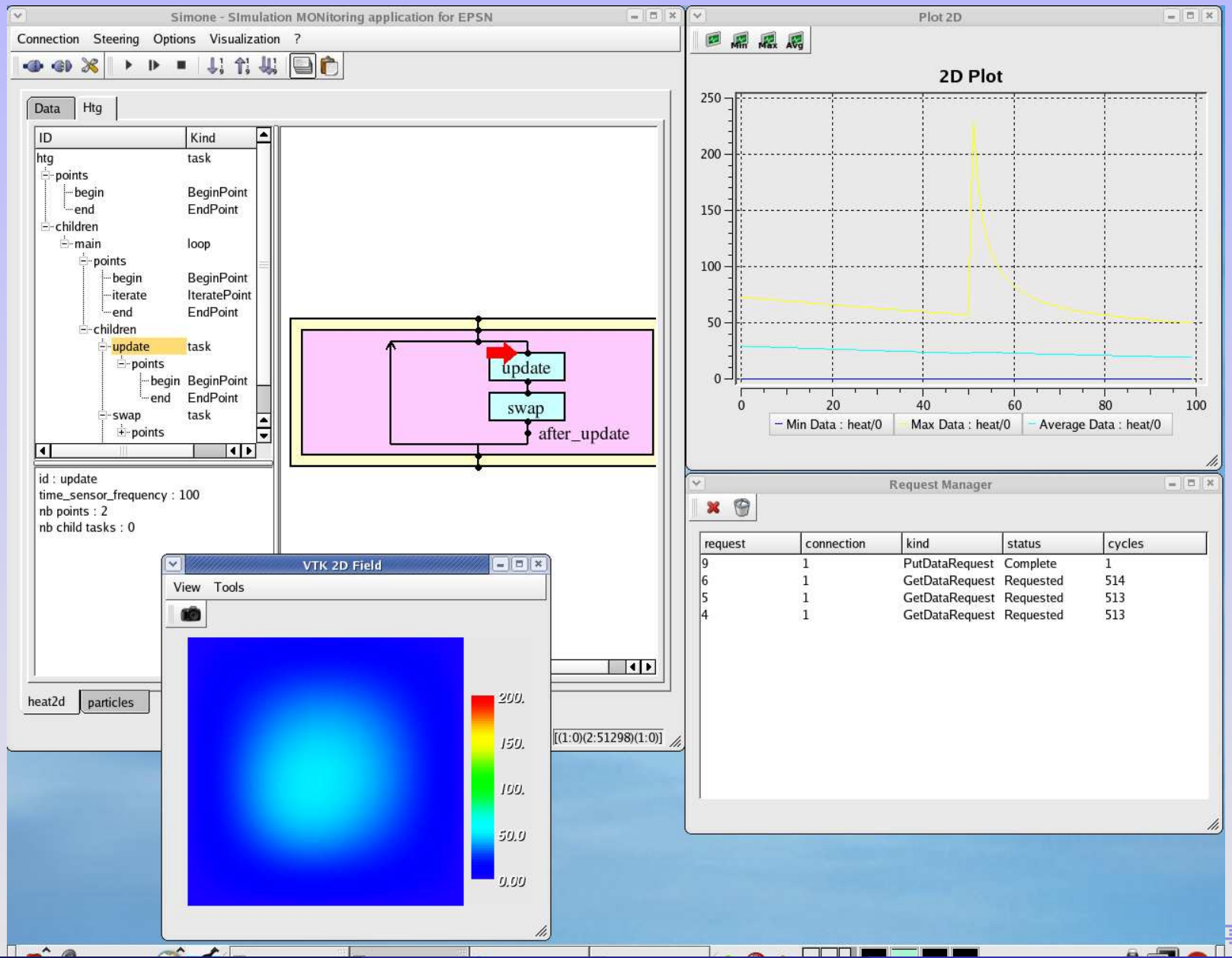
Simone


Simone - SImulation MONitoring application for EPSN
Connection Steering Options Visualization ?
Data Htg
ID Kind
htg task
points
begin BeginPoint
end EndPoint
children
main loop
points
begin BeginPoint
iterate IteratePoint
end EndPoint
children
update task
points
begin BeginPoint
end EndPoint
swap task
points
id : update
time_sensor_frequency : 100
nb points : 2
nb child tasks : 0
update
swap
after_update
heat2d particles
[(1:0)(2:46999)(2:0)]
Plot 2D
2D Plot
250
200
150
100
50
0
0 20 40 60 80 100
Min Data : heat/0
Max Data : heat/0
Average Data : heat/0
Request Manager
request connection kind status cycles
9 1 PutDataRequest Complete 1
6 1 GetDataRequest Requested 471
5 1 GetDataRequest Requested 471
4 1 GetDataRequest Requested 471
VTK 2D Field
View Tools
200.
150.
100.
50.0
0.00

Simone


Simone - SImulation MONitoring application for EPSN
Connection Steering Options Visualization ?
Data Htg
ID Kind
htg task
points
begin BeginPoint
end EndPoint
children
main loop
points
begin BeginPoint
iterate IteratePoint
end EndPoint
children
update task
points
begin BeginPoint
end EndPoint
swap task
points
id : update
time_sensor_frequency : 100
nb points : 2
nb child tasks : 0
update
swap
after_update
heat2d particles
[(1:0)(2:48168)]
Plot 2D
2D Plot
Min Max Avg
250
200
150
100
50
0
0 20 40 60 80 100
Min Data : heat/0
Max Data : heat/0
Average Data : heat/0
Request Manager
request connection kind status cycles
9 1 PutDataRequest Complete 1
6 1 GetDataRequest Requested 482
5 1 GetDataRequest Requested 482
4 1 GetDataRequest Requested 482
VTK 2D Field
View Tools
200.
150.
100.
50.0
0.00

Simone


Simone - SImulation MONitoring application for EPSN
Connection Steering Options Visualization ?
Data Htg
ID Kind
htg task
points
begin BeginPoint
end EndPoint
children
main loop
points
begin BeginPoint
iterate IteratePoint
end EndPoint
children
update task
points
begin BeginPoint
end EndPoint
swap task
points
id : update
time_sensor_frequency : 100
nb points : 2
nb child tasks : 0
update
swap
after_update
heat2d particles
[(1:0)(2:51298)(1:0)]
Plot 2D
2D Plot
250
200
150
100
50
0
0 20 40 60 80 100
Min Data : heat/0
Max Data : heat/0
Average Data : heat/0
Request Manager
request connection kind status cycles
9 1 PutDataRequest Complete 1
6 1 GetDataRequest Requested 514
5 1 GetDataRequest Requested 513
4 1 GetDataRequest Requested 513
VTK 2D Field
View Tools
200.
150.
100.
50.0
0.00

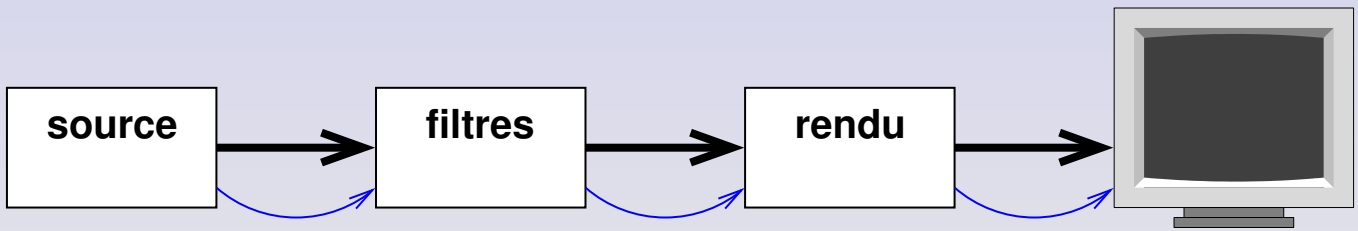
Les systèmes *Event-driven*

Principe

- Modèle *data-flow* basé sur un pipe-line classique
- Une mise-à-jour est lancée automatiquement pour chaque *évènement* dans le pipe-line
  - changement dans les connections du pipe-line
  - modification des paramètres d'un élément
  - modification des données d'une source

Problèmes

⇒ Intégration dans la boucle d'évènement



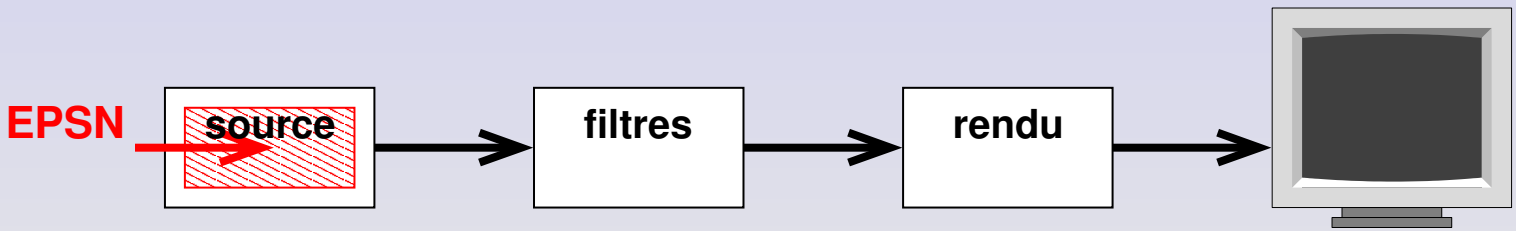
Les systèmes *Event-driven*

Principe

- Modèle *data-flow* basé sur un pipe-line classique
- Une mise-à-jour est lancée automatiquement pour chaque *évènement* dans le pipe-line
  - changement dans les connections du pipe-line
  - modification des paramètres d'un élément
  - modification des données d'une source

Problèmes

⇒ Intégration dans la boucle d'évènement



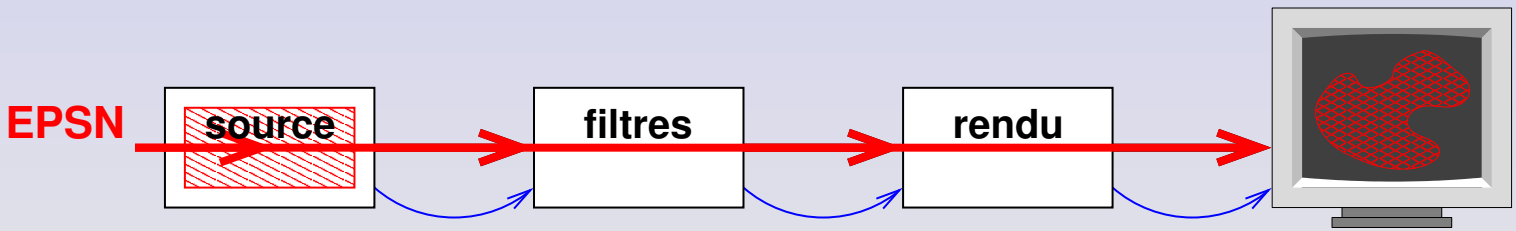
Les systèmes *Event-driven*

Principe

- Modèle *data-flow* basé sur un pipe-line classique
- Une mise-à-jour est lancée automatiquement pour chaque *évènement* dans le pipe-line
  - changement dans les connections du pipe-line
  - modification des paramètres d'un élément
  - modification des données d'une source

Problèmes

⇒ Intégration dans la boucle d'évènement



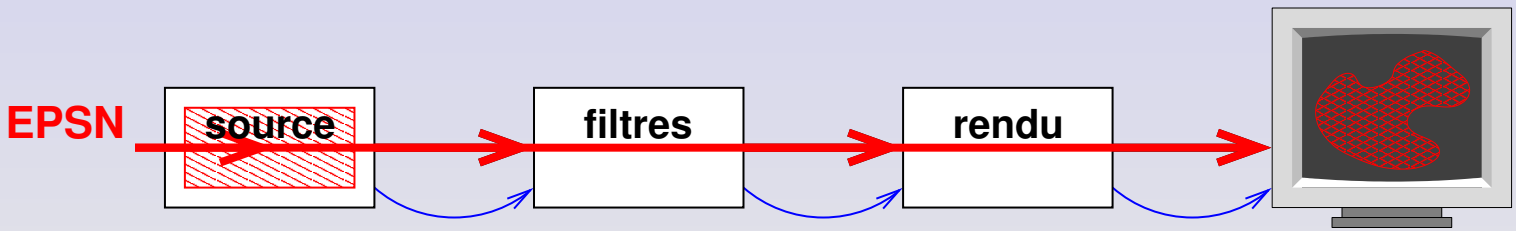
Les systèmes *Event-driven*

Principe

- Modèle *data-flow* basé sur un pipe-line classique
- Une mise-à-jour est lancée automatiquement pour chaque *évènement* dans le pipe-line
  - changement dans les connections du pipe-line
  - modification des paramètres d'un élément
  - modification des données d'une source

Problèmes

⇒ Intégration dans la boucle d'évènement



Les systèmes *Demand-driven*

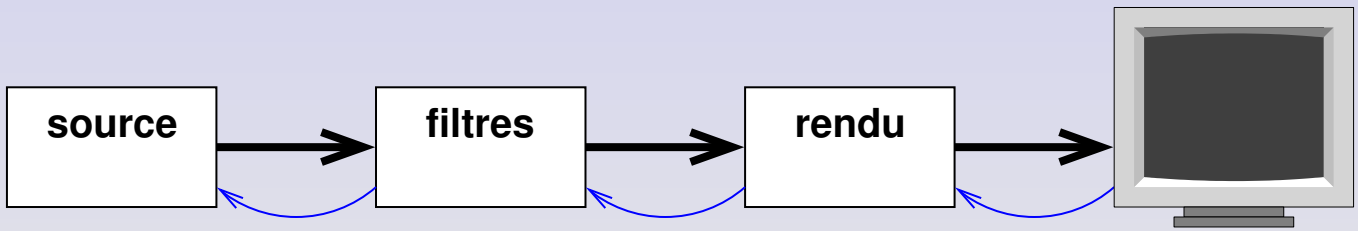
Principe

- Modèle *data-flow* basé sur un pipe-line classique
- L'utilisateur peut demander explicitement une mise-à-jour
- Les mises-à-jours automatique ne sont commandées que par les interactions utilisateur

Problèmes

⇒ Demande explicite de mise à jour

⇒ Intégration dans la boucle d'évènement



Les systèmes *Demand-driven*

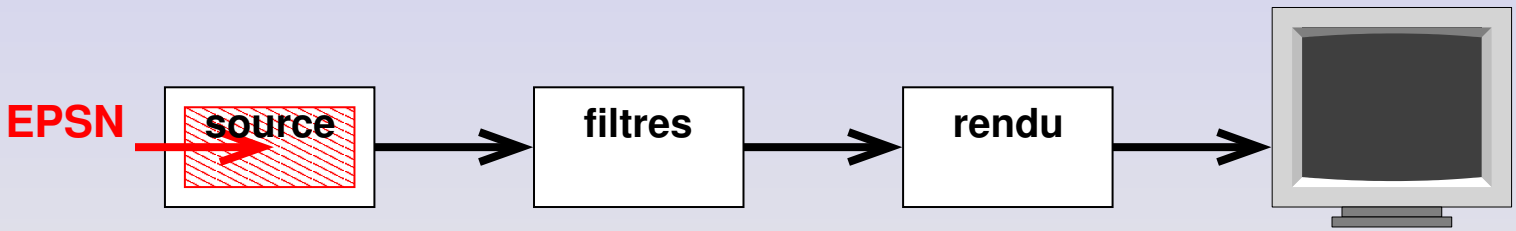
Principe

- Modèle *data-flow* basé sur un pipe-line classique
- L'utilisateur peut demander explicitement une mise-à-jour
- Les mises-à-jours automatique ne sont commandées que par les interactions utilisateur

Problèmes

⇒ Demande explicite de mise à jour

⇒ Intégration dans la boucle d'évènement



Les systèmes *Demand-driven*

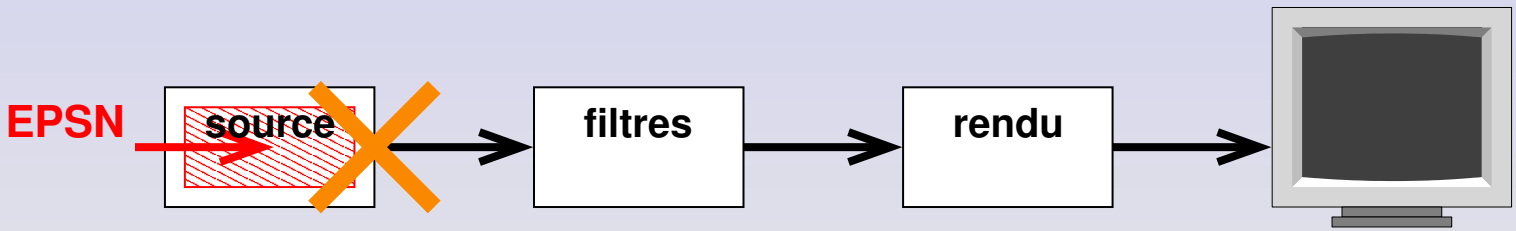
Principe

- Modèle *data-flow* basé sur un pipe-line classique
- L'utilisateur peut demander explicitement une mise-à-jour
- Les mises-à-jours automatique ne sont commandées que par les interactions utilisateur

Problèmes

⇒ Demande explicite de mise à jour

⇒ Intégration dans la boucle d'évènement



Les systèmes *Demand-driven*

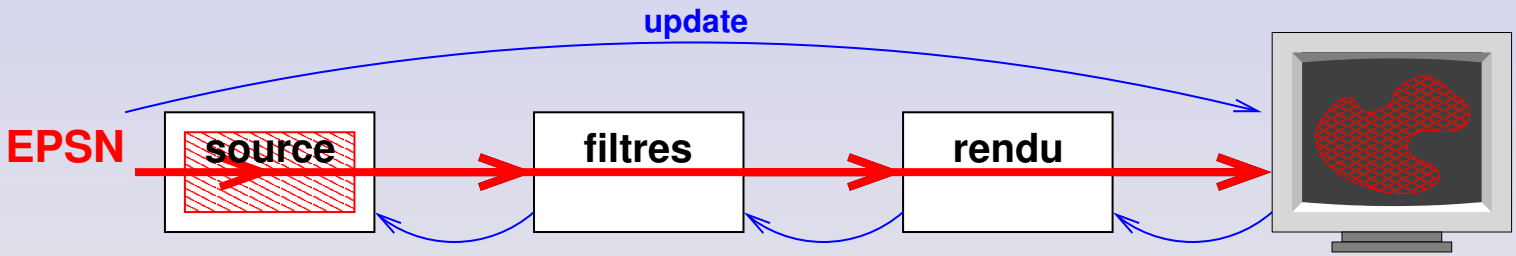
Principe

- Modèle *data-flow* basé sur un pipe-line classique
- L'utilisateur peut demander explicitement une mise-à-jour
- Les mises-à-jours automatique ne sont commandées que par les interactions utilisateur

Problèmes

⇒ Demande explicite de mise à jour

⇒ Intégration dans la boucle d'évènement



Les systèmes *Demand-driven*

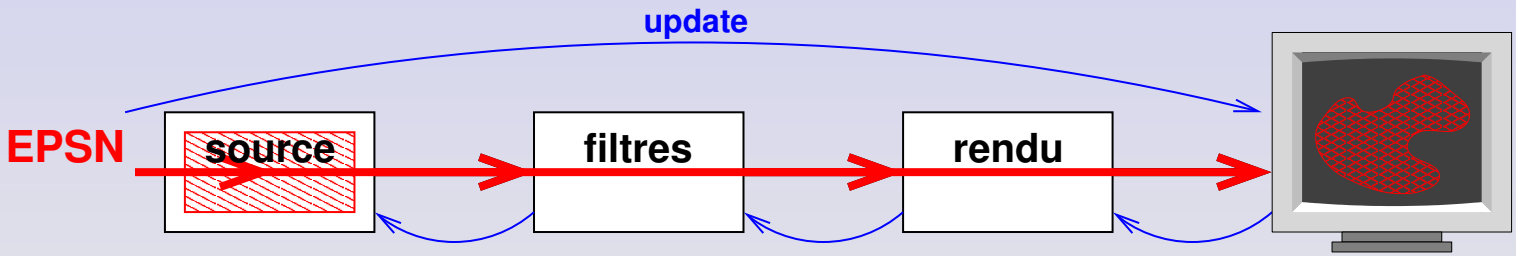
Principe

- Modèle *data-flow* basé sur un pipe-line classique
- L'utilisateur peut demander explicitement une mise-à-jour
- Les mises-à-jours automatique ne sont commandées que par les interactions utilisateur

Problèmes

⇒ Demande explicite de mise à jour

⇒ Intégration dans la boucle d'évènement



Intégration à la boucle d'évènements

Intégration à la boucle d'évènements

- Accès sérialisé au données de la source (lock/unlock)
- Signalement des modifications (modified)

Demande explicite de mise à jour

- Les systèmes de visualisation sont rarement *thread-safe*
- Utilisation d'évènements spécifique au système
- Utilisation d'un *timer* ou d'une fonction *idle*

Source de données pour VTK

RedSYM vs. VTK

- Même système de données
  - objets complexes avec des séries de données associées
  - séries $\leftrightarrow$ listes de *tuples* à plusieurs composantes homogènes
- Partage de la mémoire ou recopie
- Equivalence :
  - champs $\longrightarrow$ `vtkImageData`
  - points $\longrightarrow$ `vtkPoints`
  - maillages $\longrightarrow$ `vtkUnstructuredGrid`
  - séries $\longrightarrow$ `vtkDataArray`

En cours

$\Rightarrow$ Ecritures de sources RedGRID pour VTK

Source de données pour VTK

RedSYM vs. VTK

- Même système de données
  - objets complexes avec des séries de données associées
  - séries $\leftrightarrow$ listes de *tuples* à plusieurs composantes homogènes
- Partage de la mémoire ou recopie
- Equivalence :
  - champs $\longrightarrow$ `vtkImageData`
  - points $\longrightarrow$ `vtkPoints`
  - maillages $\longrightarrow$ `vtkUnstructuredGrid`
  - séries $\longrightarrow$ `vtkDataArray`

En cours

$\Rightarrow$ Ecritures de sources RedGRID pour VTK

Intégration dans un système de visualisation parallèle

Principe

- Architecture maître / esclaves
- Données distribuées
- Pipe-lines parallèles