

Environnements pour la visualisation de simulations scientifiques interactives sur architectures découplées

Emmanuel Melin



Systemes de RV



la station de
calculs

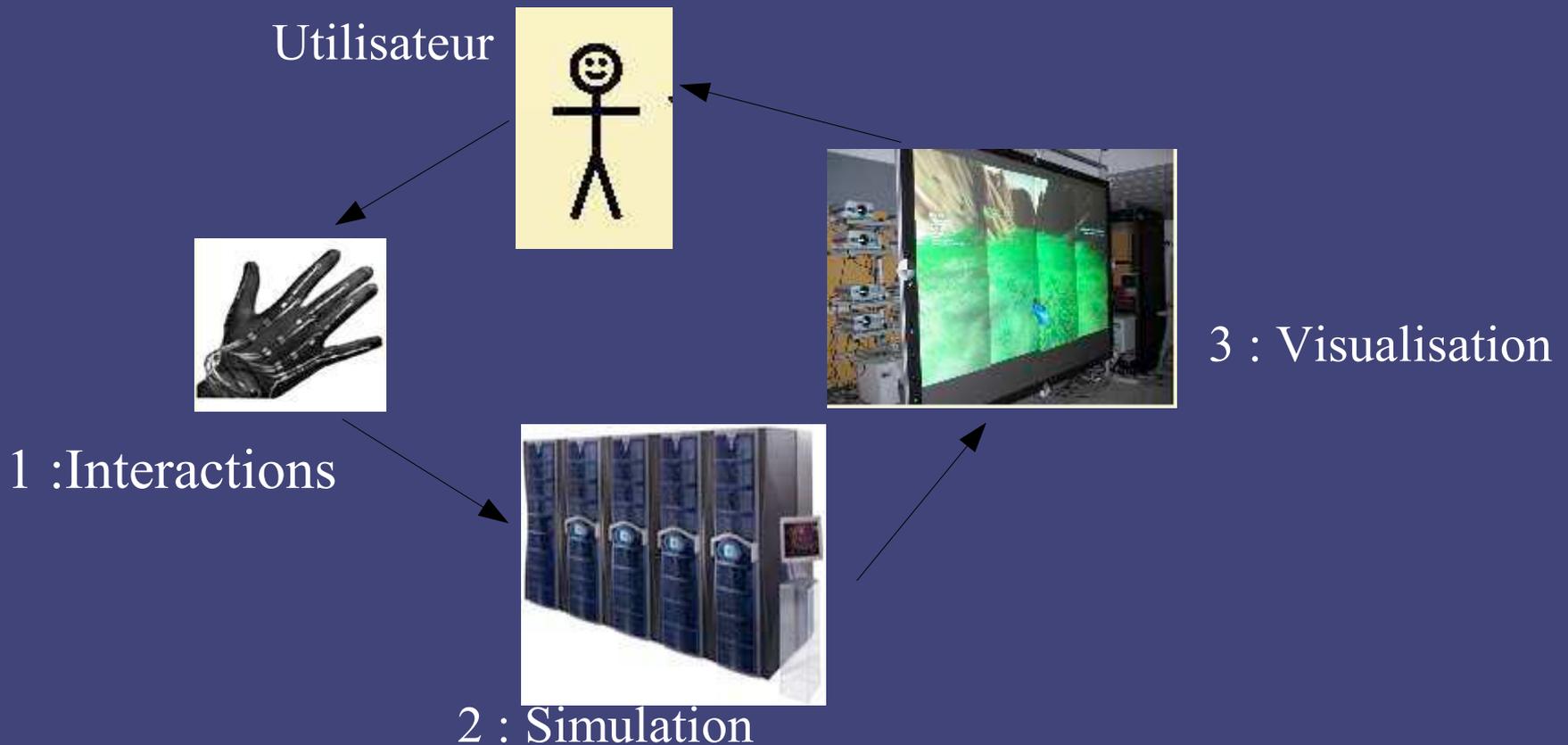


écrans de
visualisation

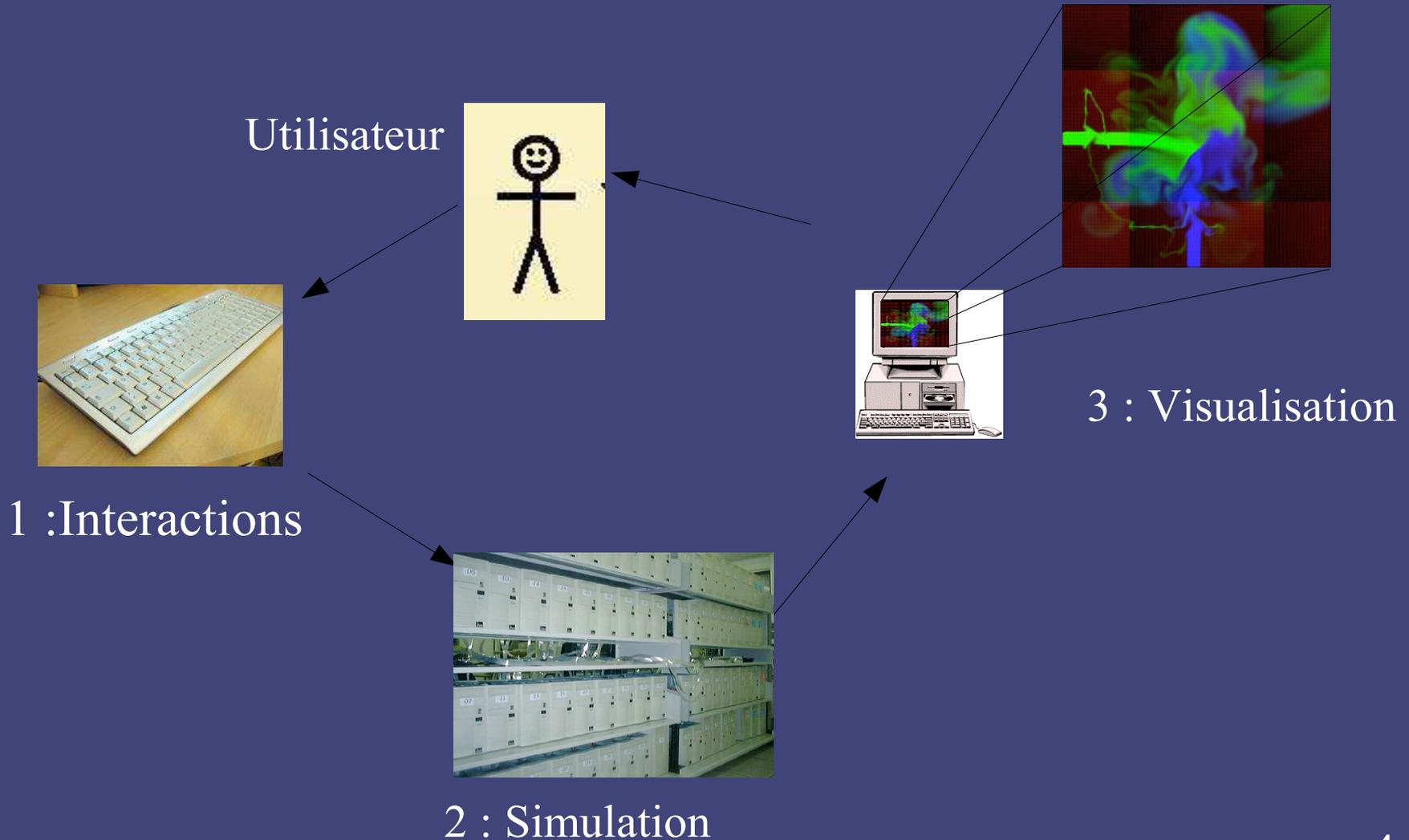


périphériques
d'entrées

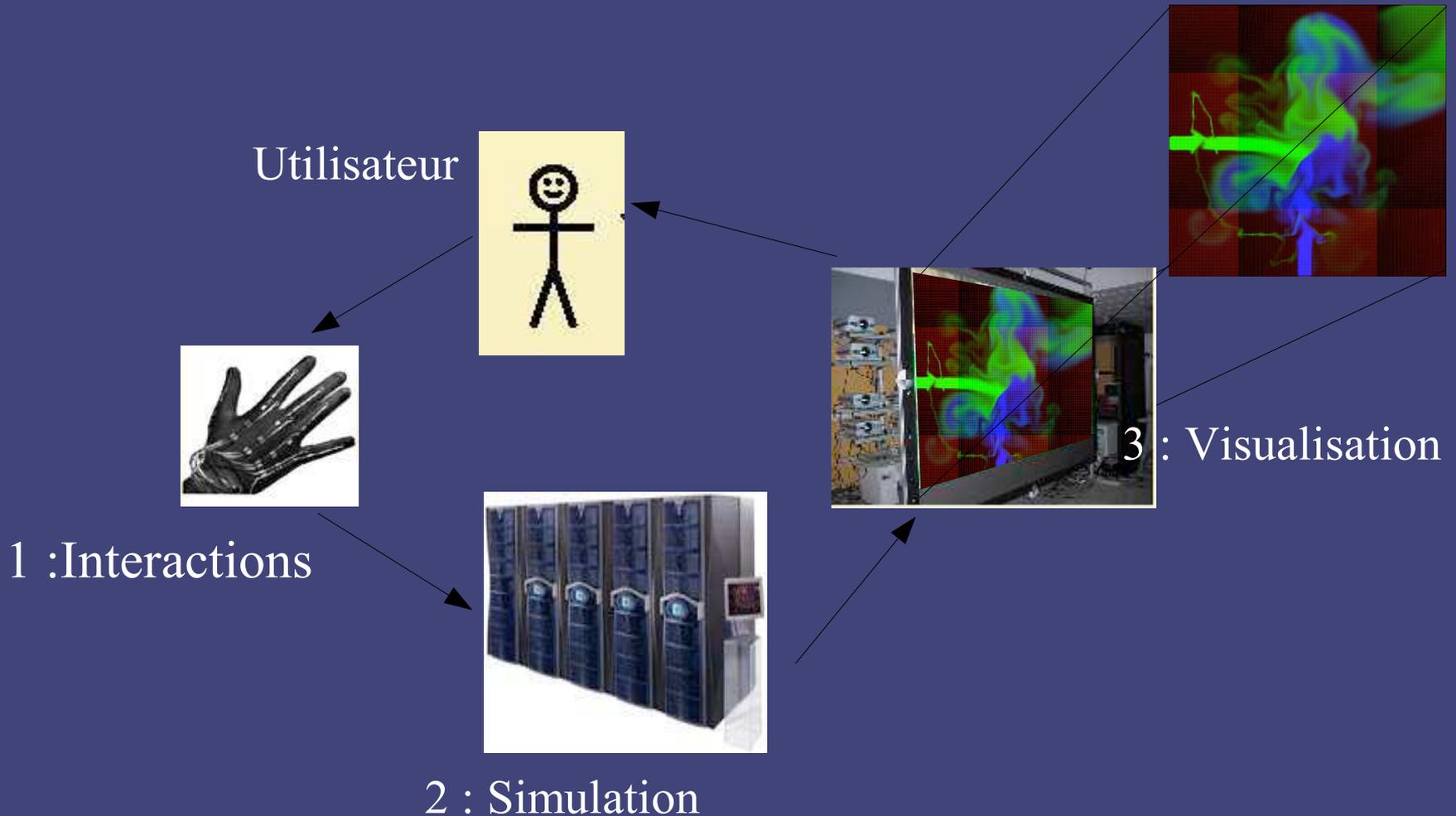
Fonctionnement itératif d'une application de RV



L'utilisation itérative d'une application de simulation



Simulation scientifique dans un environnement de RV

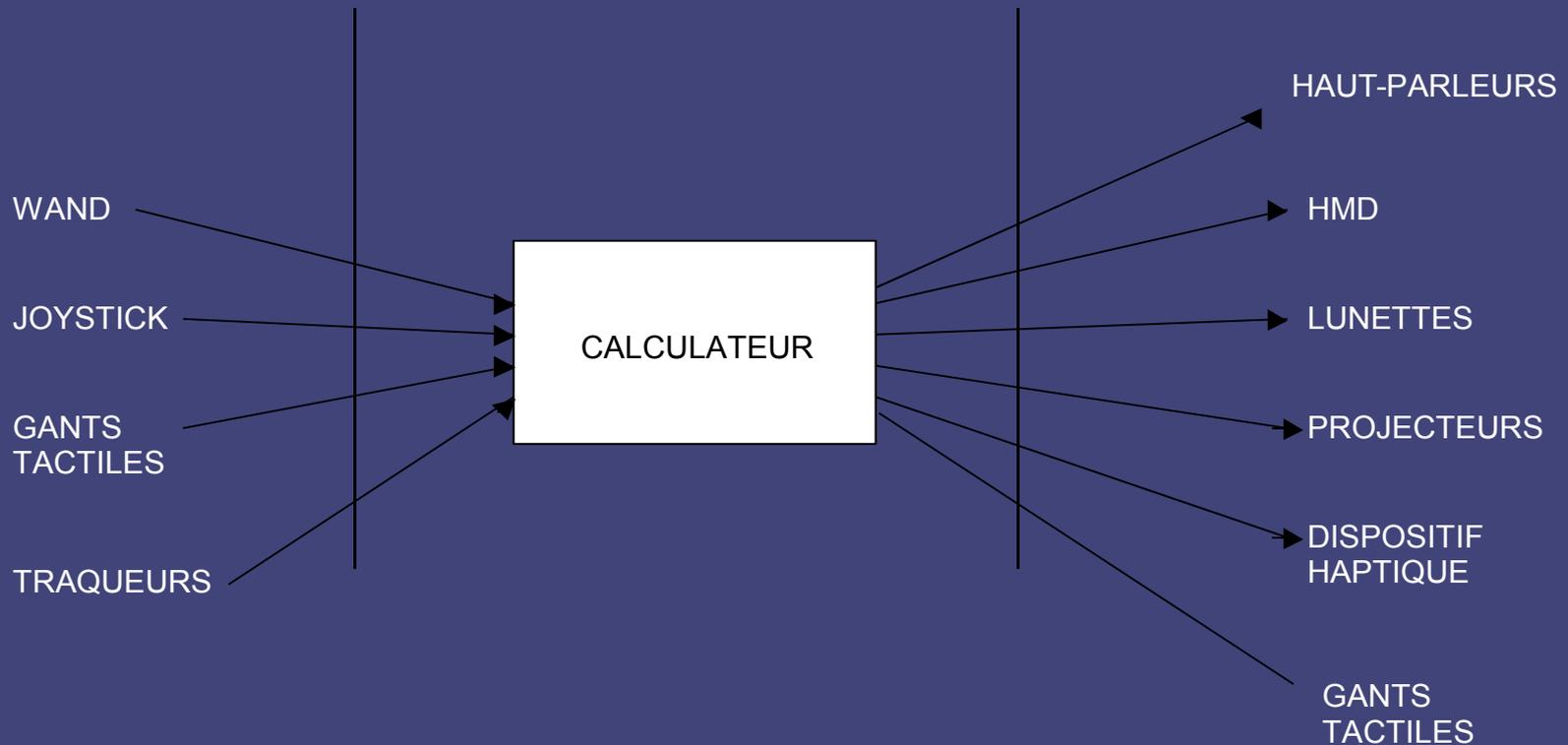


Systemes de réalité virtuelle

INTERFACES D'ENTREES

CALCULATEUR

INTERFACES DE SORTIE



- **Machine puissante avec éventuellement plusieurs sorties vidéos**

Systemes de réalit  virtuelle + Simulation

INTERFACES D'ENTREES

CALCULATEUR

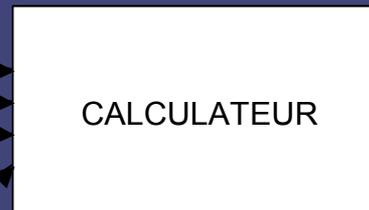
INTERFACES DE SORTIE

WAND

JOYSTICK

GANTS
TACTILES

TRAQUEURS



HAUT-PARLEURS

HMD

LUNETTES

PROJECTEURS

DISPOSITIF
HAPTIQUE

GANTS
TACTILES

Simulation

Systemes de r ealit e virtuelle: le calculateur

Un exemple : le SGI ONYX

Multiprocesseur, multipipe, multicanal

**Machine haute performance mais non
extensible, co t  lev e (~ 500.000\$)**



Onyx 3000

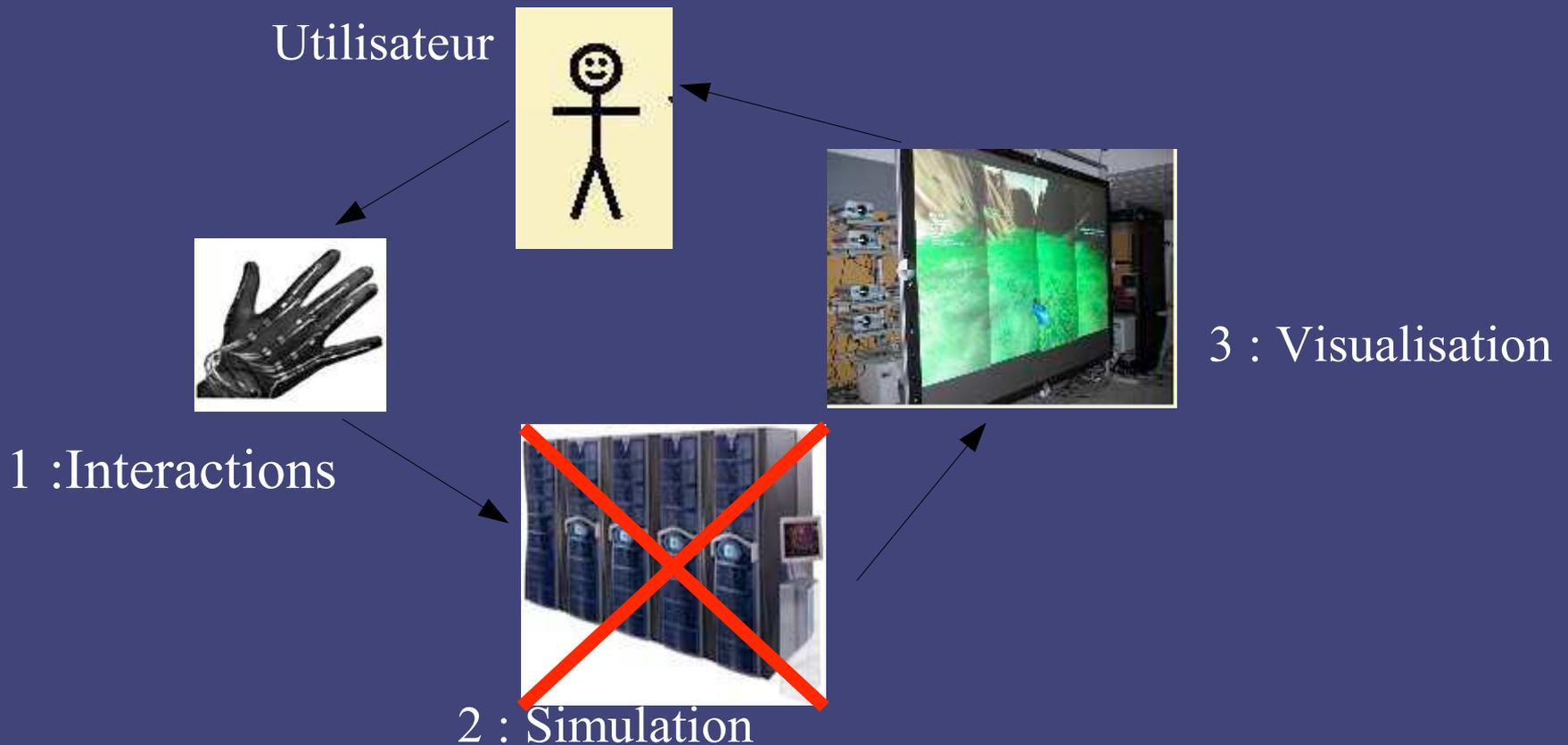
Los Alamos Koolaid Cluster



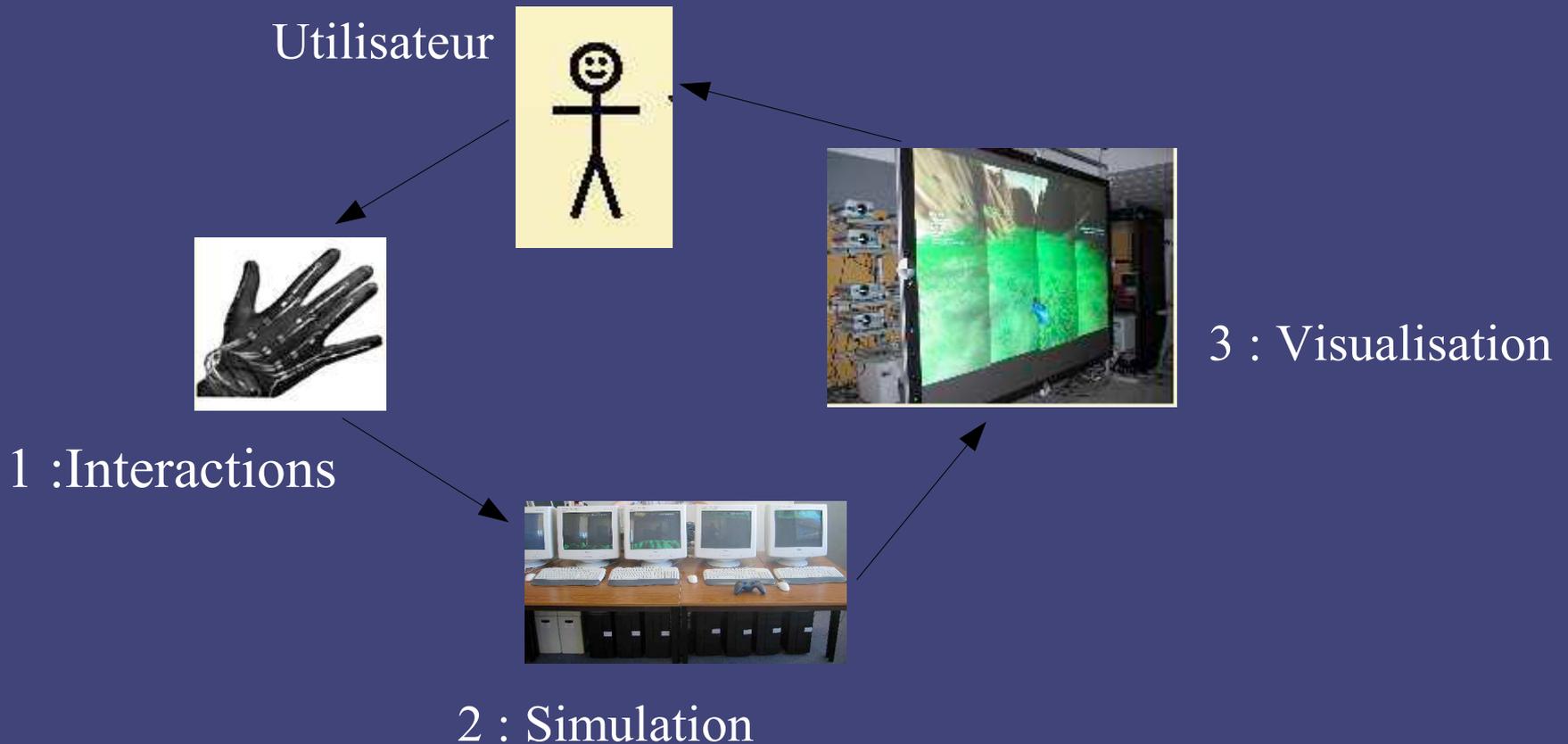
Lifo Black Cluster



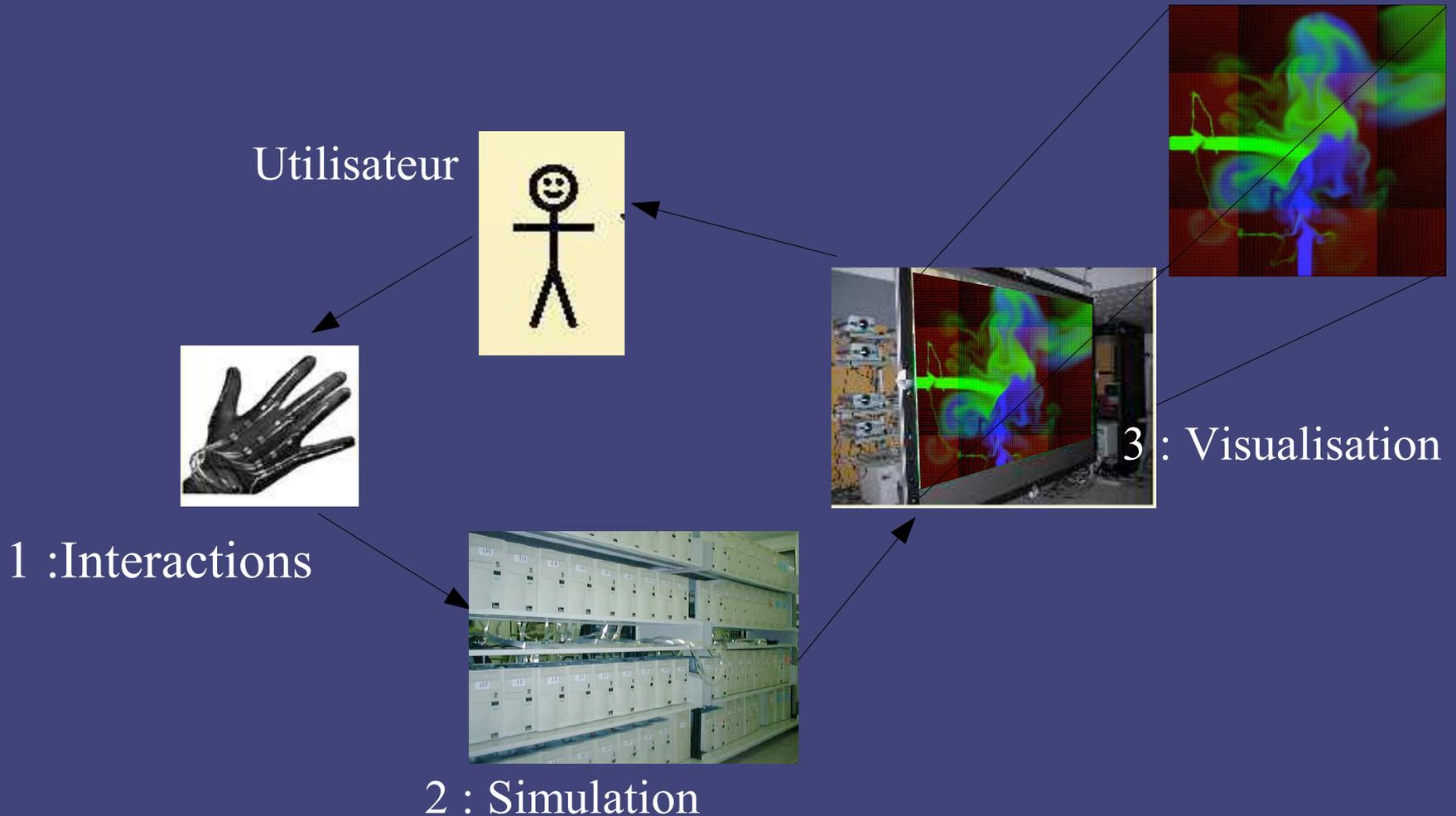
Changeons de machine de RV!



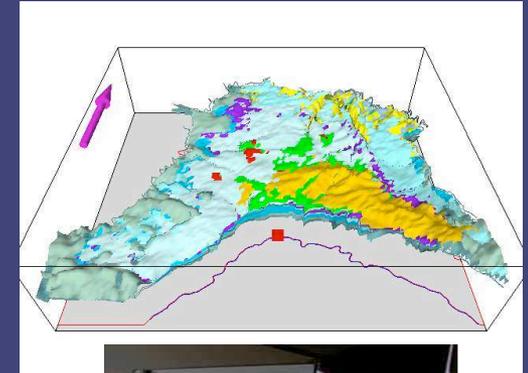
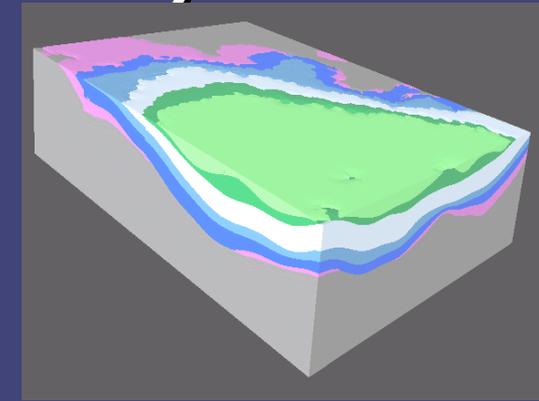
RV avec une grappe de PC



Simulation scientifique dans un environnement de RV



Le Visio Plan (Workbench)



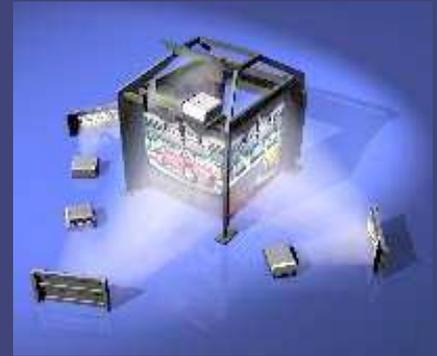
**Visite de la ville de Bruxelles sur le
Workbench du BRGM**

Objectif général du thème RV

- Solutions logicielles pour piloter des environnements multi-projecteurs par des grappes de PC : *Coût, Standard, Extensible, Modulaire*
- Optimisation du couplage rendu (affichage) / pré-rendu (simulation) sur une grappe de PC



Grappes de PC et Réalité Virtuelle



Les points durs : 3 niveaux de sync :

- 1. Datalock : cohérence des images**
- 2. Swaplock : sync échange de buffers**
- 3. Genlock : sync signaux vidéo**



GenLock

- Stabilité des images
- Stéréo active
- Solutions :



1. Hardware : Wildcat 3Dlabs, NV quadro FX
2. Hard/Soft : Artabel, Orad, MetaVR
3. Software : **SoftGenLock** (LIFO, ID)

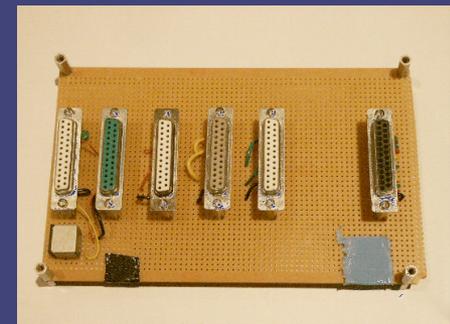
Maximiser l'utilisation de matériel/logiciel standard pour minimiser les dépendances avec la carte graphique.



Pérenniser les développements

SoftGenLock + Réseau

- **Communications nécessaires pour mesurer T_{ref}**
- **Besoins :**
 - Temps-réel
 - Accessible depuis une tâche RTAI ou RT-Linux
 - 1 bit en émission depuis la machine de référence
- **Solution: Réseau basé sur le port parallèle**
 - Simple à construire (pas d'électronique)
 - Bon marché (50 euros)
 - Très rapide : 3-5 us
 - Port // contrôlable par RTAI/RT-Linux

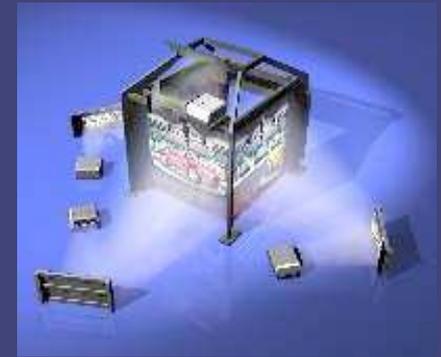


Grappes de PC et Réalité Virtuelle



Les points durs : 3 niveaux de sync :

- 1. Datalock : cohérence des images**
- 2. Swaplock : sync échange de buffers**
- 3. Genlock : sync signaux vidéo**



Net Juggler



Tracker events

Configuration events

Wand events

Net Juggler communication layer (data lock)

VJ Application

VJ Application

VJ Application

Net Juggler

swaplock



Communications Net Juggler

- **Données échangées (événements d'entrée et de config):**
 - **Faible volume**
 - **Indépendant de la complexité graphique de la scène**
- **Implémentation de la couche de communication:**
 - **Regroupement de données**
 - **Communications collectives optimisées**

Passage à l'échelle

Faible bande passante

Fast-ethernet est suffisant



Grappes de PC et Visualisation

Visualisation

- Genlock : sync signaux vidéo
 ➔ softgenlock.sourceforge.net
- Swaplock : sync échange de buffers
 ➔ netjuggler.sourceforge.net
- Datalock : cohérence des images
 ➔ netjuggler.sourceforge.net

Parallélisation

- Parallélisation du rendu
 ➔ netjuggler.sourceforge.net
- Parallélisation du prérendu
 ➔ ?

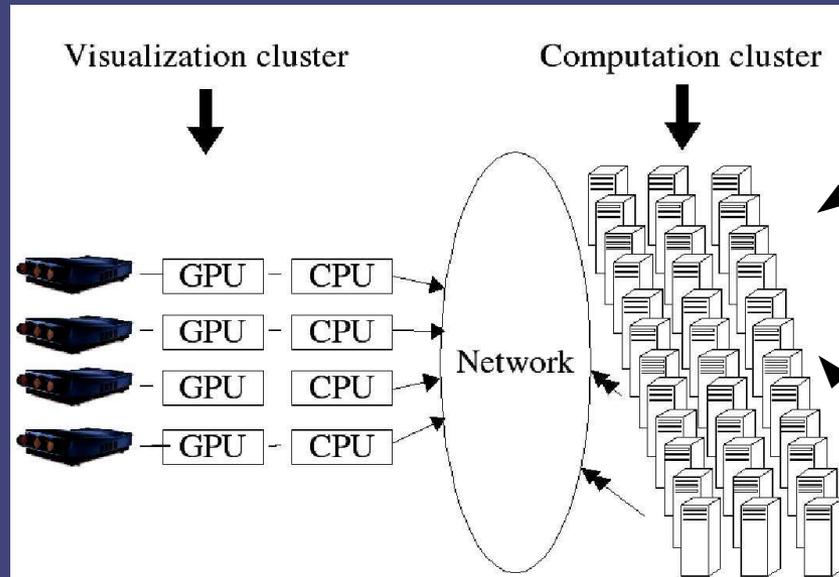
Un environnement réel sous Netjuggler et Softgenlock

Effet de flou: stereo active



Visite de la ville de Bruxelles sur le Workbench du BRGM

Au-delà du rendu graphique parallèle ?



- Rendu audio
- Rendu haptique
- Tracking
- Capture de mouvement
- Traitement Vidéo

- Simulations:
 - Mondes synthétiques
 - Physique

- Éviter le goulot d'étranglement du séquentiel
- Coordonner du code hétérogène (parallèle)
- Garantir la performance (latence, niveau de détail)
- Garantir la cohérence

→ Besoin d'un intergiciel de Réalité virtuelle ?

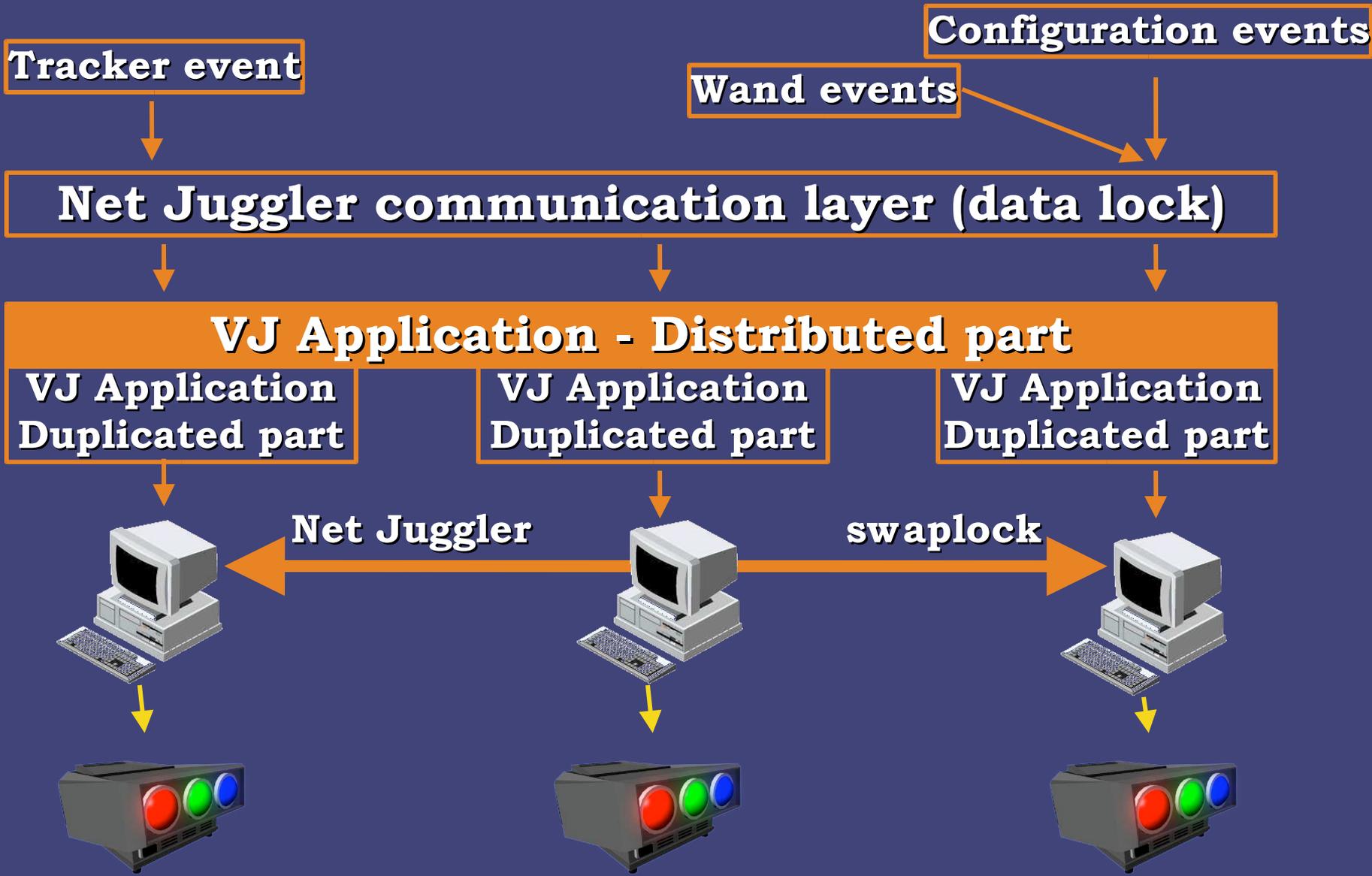
Le couplage avec Net Juggler Démonstrateur I

- **Simulation interactive d'écoulement de fluide**
- **Solveur Navier-Stoke // (PETSC)**
- **Visualisation multi-projecteurs**
- **Mélange interactif des fluides**

1. Solveur séquentiel (dupliqué) : 8 fps

2. Solveur parallèle sur 6 machines : 25 fps

Net Juggler



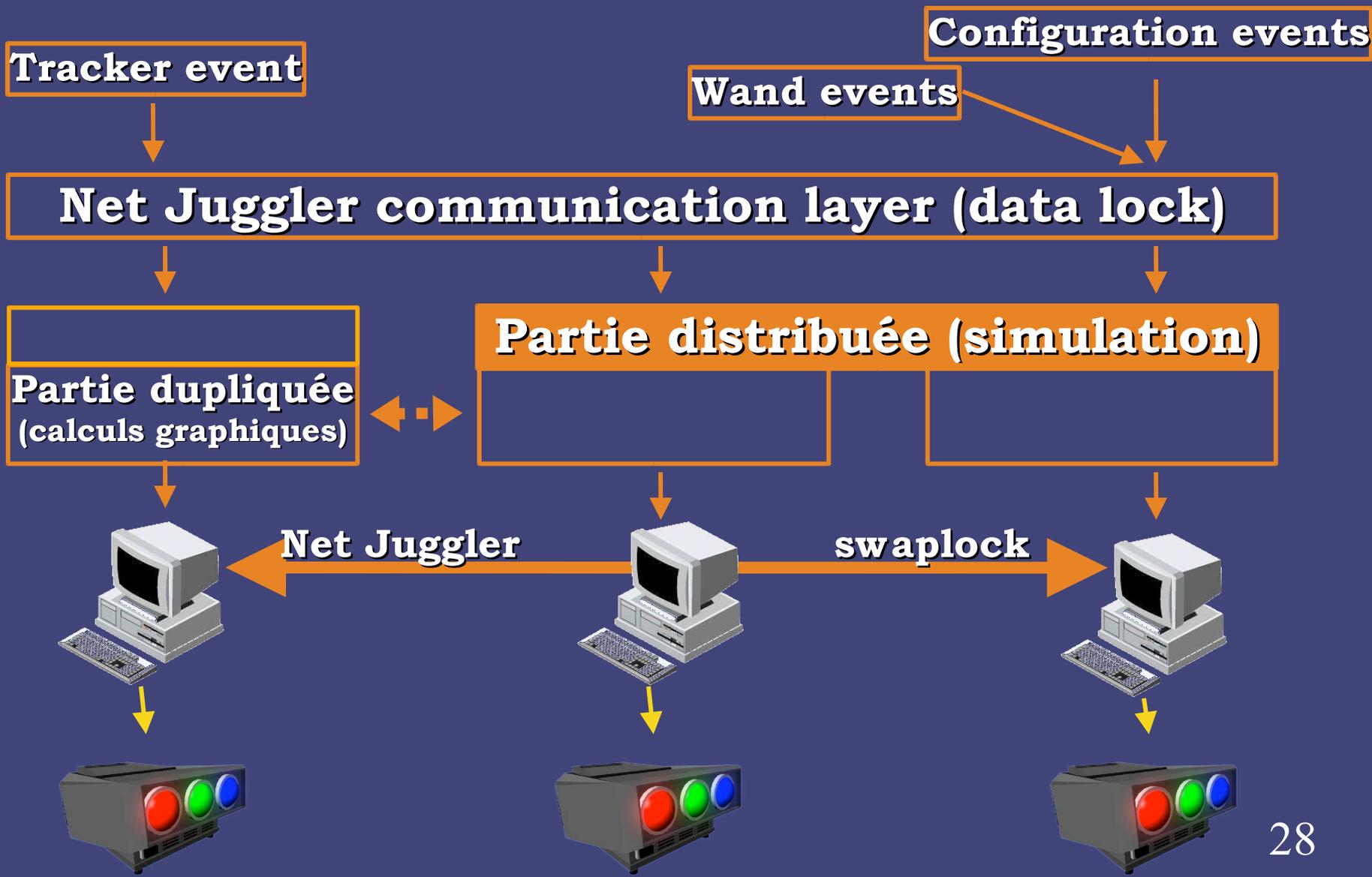
Le couplage avec Net Juggler

Démonstrateur II

- **Simulation des fluides à base d'un solveur multi-grid**
- **Découplage plus important de la simulation et de l'affichage**
- **Introduction d'une couche de pré-traitement des résultats de la simulation pour le rendu graphique**

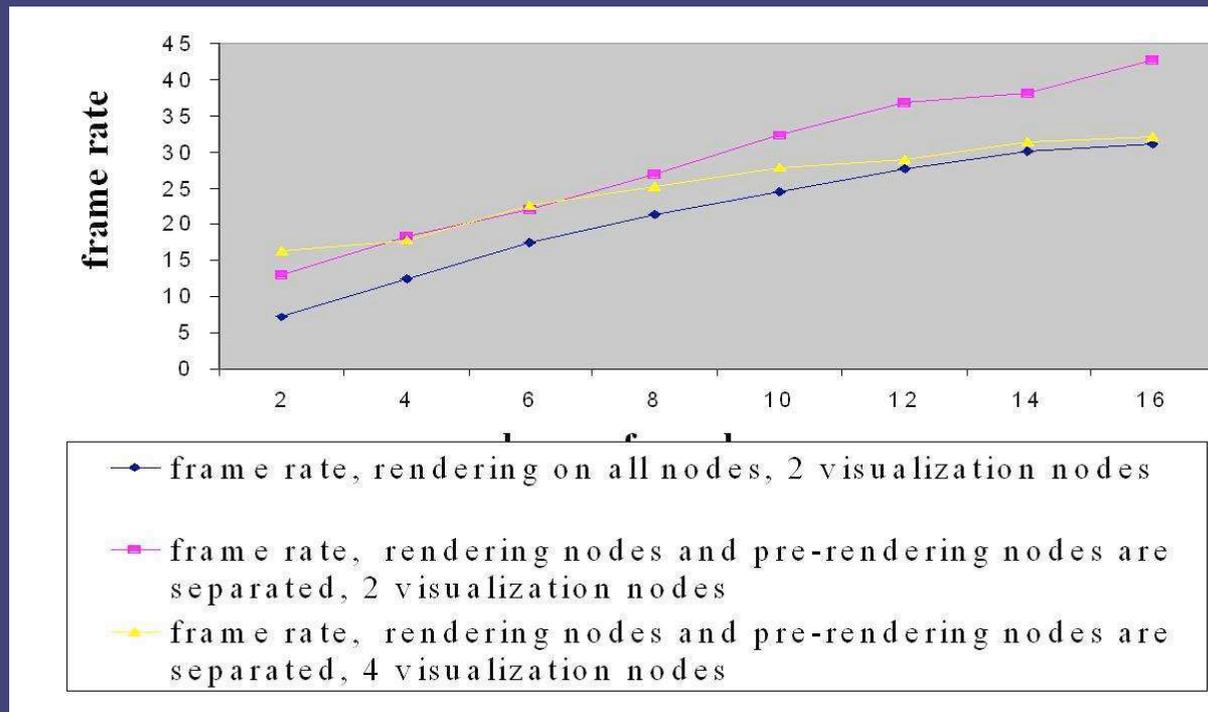
Le couplage avec Net Juggler

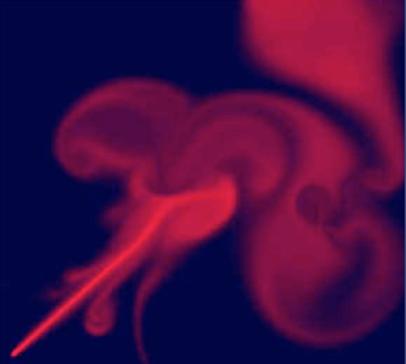
Démonstrateur II



Quelques résultats

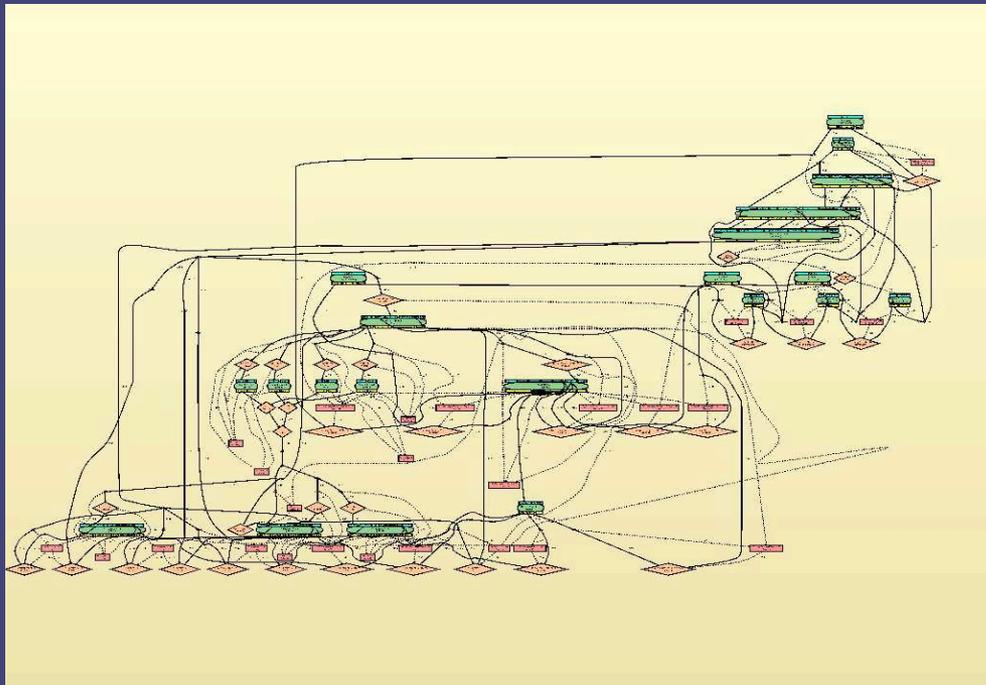
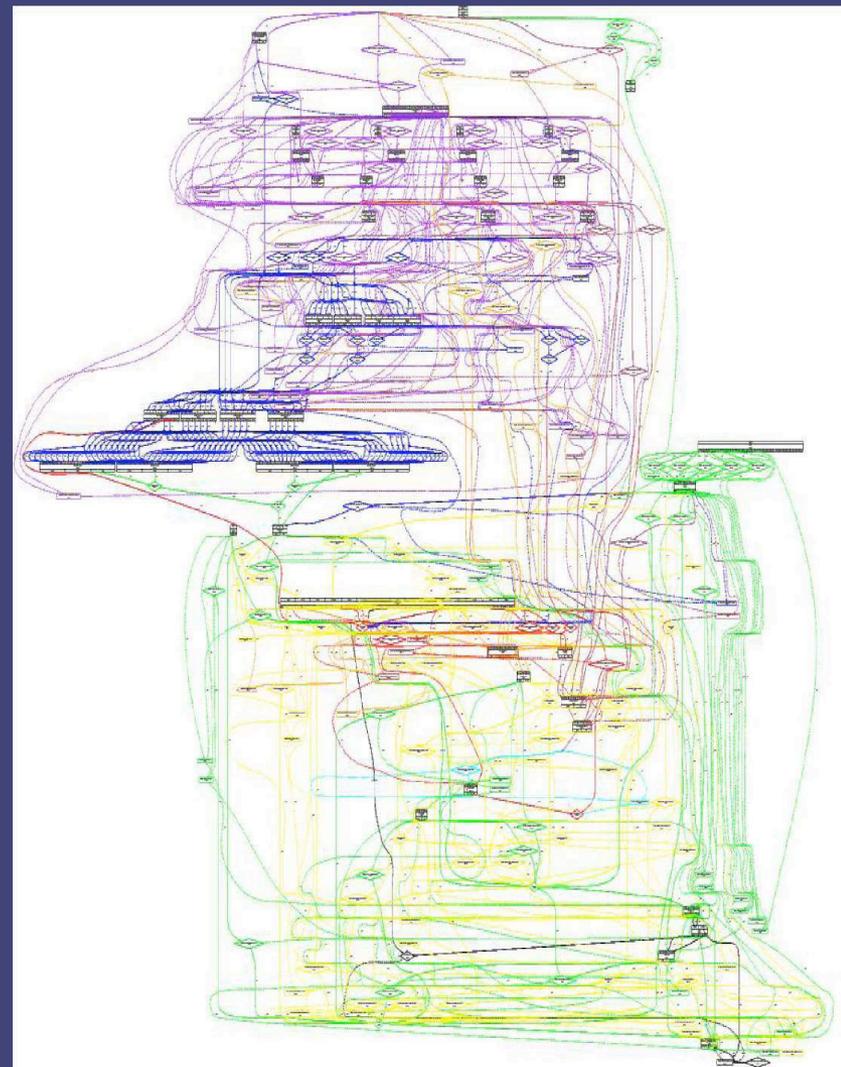
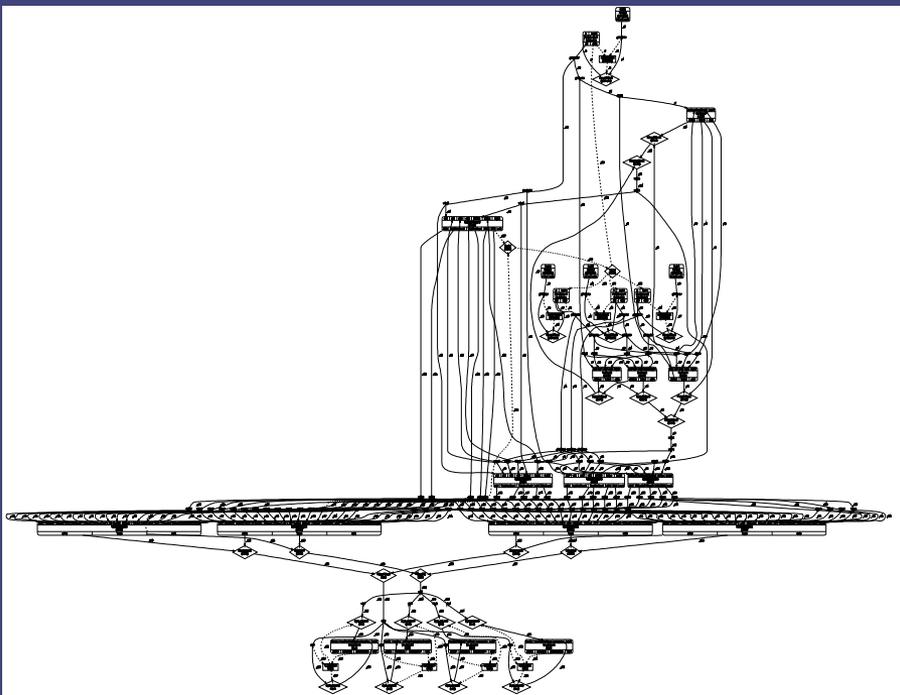
- **Cluster 8 nœuds**
 - **bi-Xeon 2.6GHz**
- **Cartes graphiques**
 - **GE-Force FX5900**
- **Réseau**
 - **Gigabit Ethernet**
- **Taille de la grille**
 - **160x160**





Bilan

- **Couplage simulation/visu possible dans NetJuggler**
- **Résultats encourageants en terme de passage à l'échelle**
- **Couplage encore fortement dépendant de l'affichage (Une étape de solveur par frame)**
- **Pas de support pour la mise au point (niveau de détail, volume de données, latence, synchronisations complexes etc...)**



**4 cameras,
50 processeurs,
16 projecteurs.**

Programmation parallèle classique: MPI, Open MP , PVM

- ✓ **Haute performance**
- ✓ **Des communications collectives**
- ✗ **Communication seulement FIFO**
- ✗ **Des calculs et des communications entrelacés**
 - Couplage de code difficile

Intergiciel d'intégration d'exécutifs communicants: CORBA, Padico™

- ✓ **Couplage de codes**
- ✗ **Communication seulement FIFO**
- ✗ **Surcoût à l'exécution**
- ✗ **Communications collectives limitées**

Intergiciels de visualisation

VTK, OpenDX, Amira

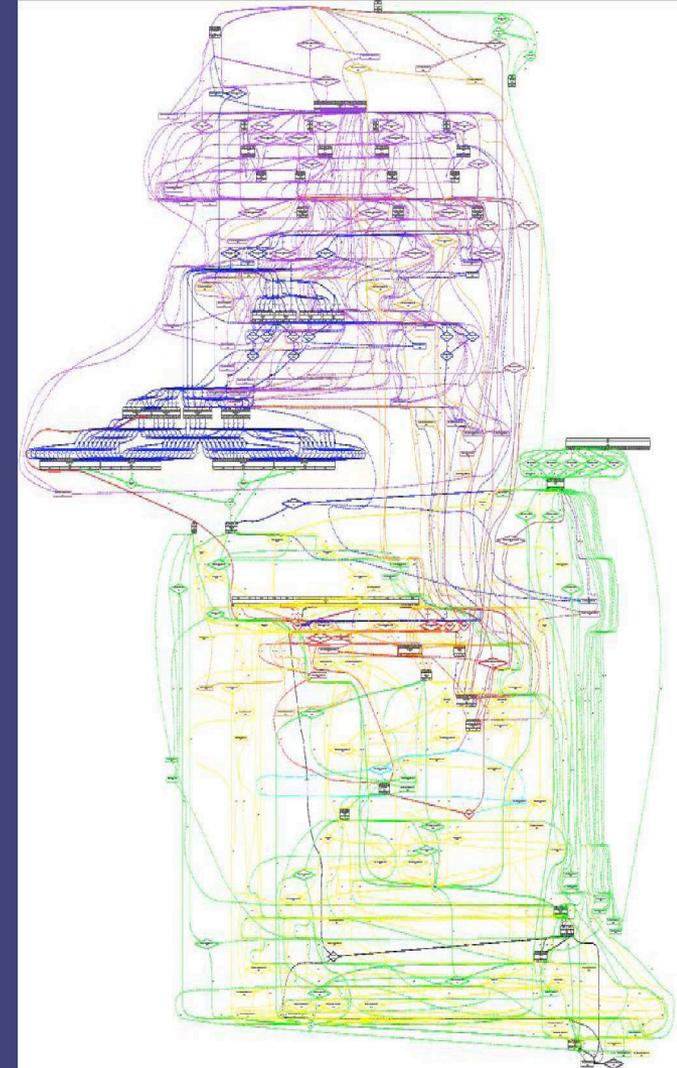
A partir des données distribuées sur la grappe:

- ✓ Parallélisme de donnée
- ✓ Parallélisme de rendu (sort-last)
- ✓ Design modulaire et data-flow (flux de données)
- ✗ Limité à la visualisation

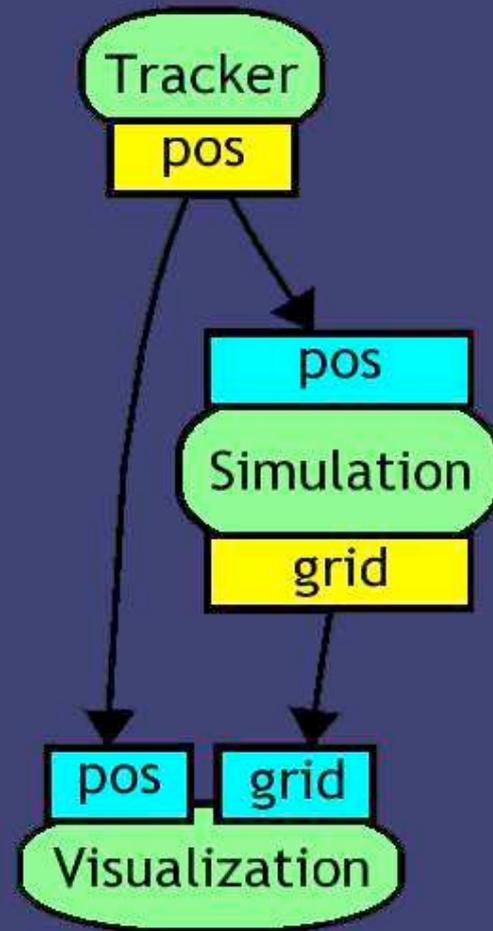
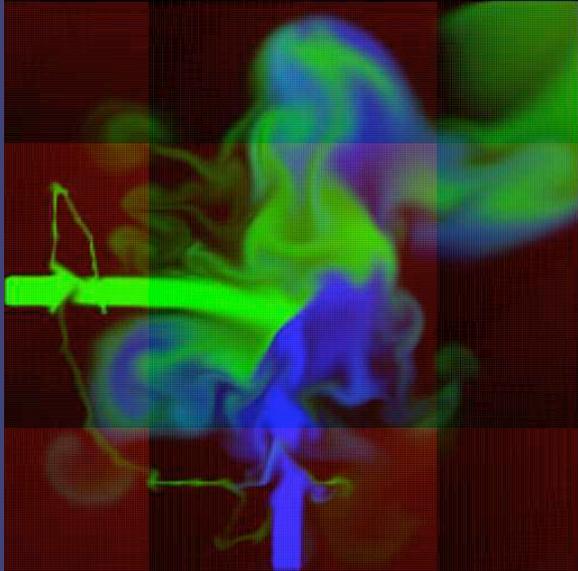
Intergiciels réalité virtuelle: Vrjuggler, Netjuggler, OpenMask

- ✓ rendu multi-écrans
- ✓ gestion des périphériques via le sampling
- ✗ Pas de couplage de code parallèle

- Une application= Modules + Réseau
- Modules: Une boucle de calcul (typiquement un processus)
- Réseau:
 - un graphe de flux de données (dataflow)
 - Spécification des liaisons entre modules
 - Spécification de la façon dont sont gérés les messages:
 - Filtres: implantation des communications collectives
 - Synchroniseurs: implantation de la politique de couplage

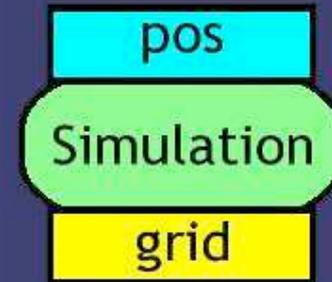
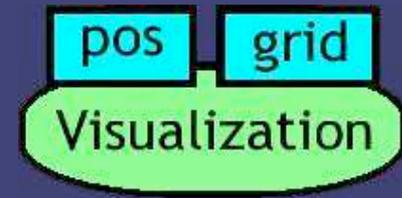


FlowVR: Un exemple



Le module FlowVR

- **Boucle sans fin**
- **Consommateur/producteur**
- **Pas de connaissance des autres modules**
 - Réutilisable
 - Indépendant des politiques de couplage
- **Pour chaque itération:**
 - Un message est lu sur chaque port d'entrée
 - Un message peut être écrit sur chaque port de sortie



Un code flowVR

Méthodes principales de l'API FlowVR :

init : définition des ports du module

wait : attend l'itération suivante

get : récupère le 1er message sur le port indiqué

put : envoie un message sur le port indiqué (1 message / itération)

Condition pour qu'une nouvelle itération débute : chaque port en entrée doit avoir (au moins) un message disponible.

Boucle du module de Simu :

tant que en cours d'exécution

wait()

get(pos)

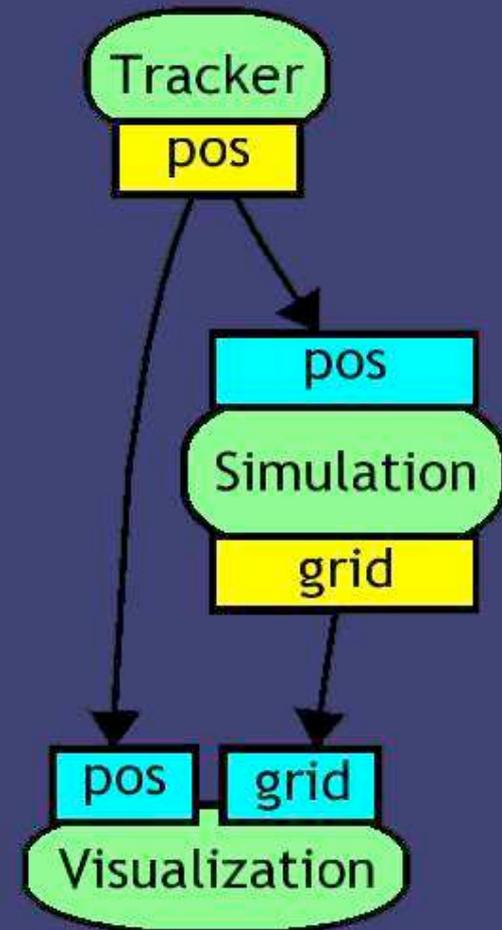
// calculs ...

put(donnees)



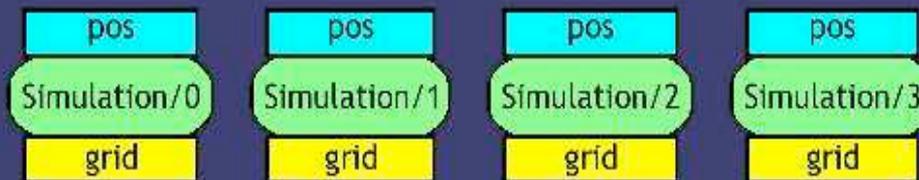
Les connexions

- **Connections des entrées sur les sorties**
- **Communications FIFO par défaut**



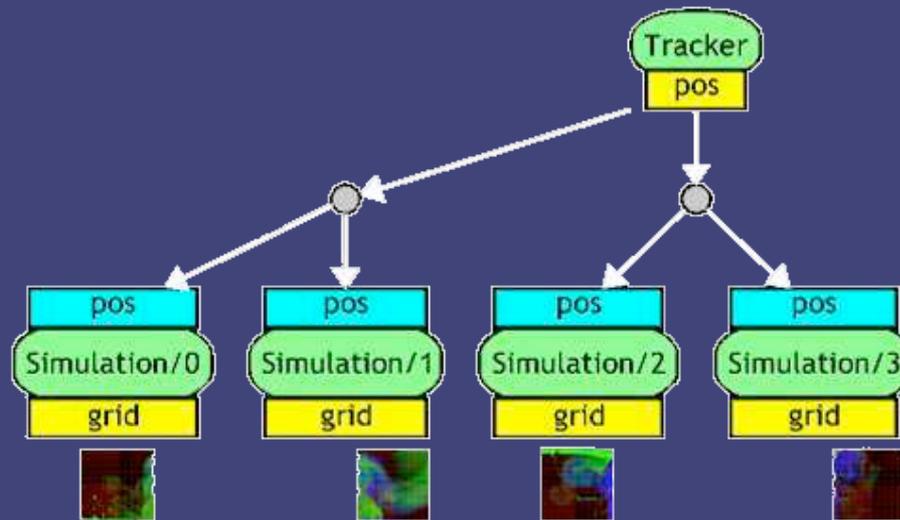
Couplage de code parallèle

- Les codes natifs MPI sont transparents pour FlowVR
- Le point de vue FlowVR:
 - 1 module par processus



Couplage de code parallèle

- **Le point de vue FlowVR:**
 - Les entrées doivent être distribuées ou réparties
 - Chaque module peut gérer tout ou partie des données
 - Il peut être nécessaire de rassembler les données



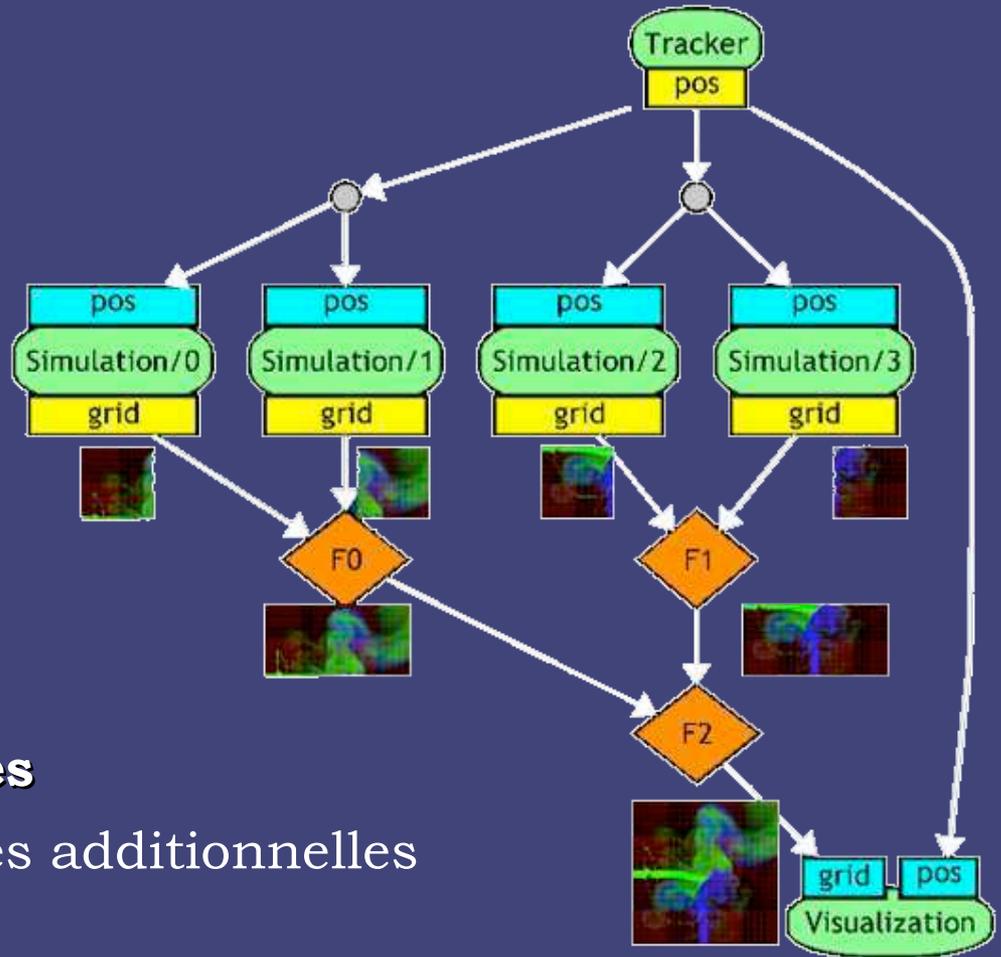
Les filtres

- **Gestion des messages:**

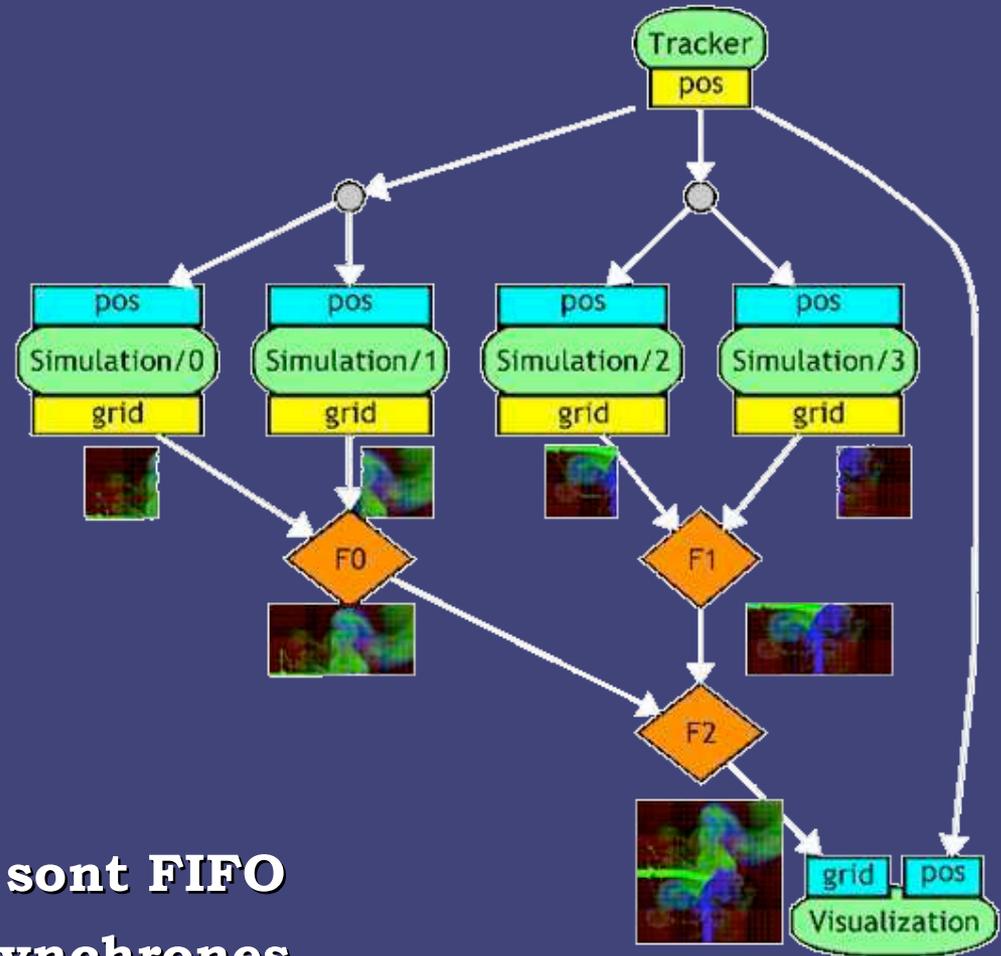
- Scatter
- Gather
- Filtrage
- Conversion
- Compression

- **Ils utilisent des estampilles**

- Informations sémantiques additionnelles
 - sources
 - numérotation
 - coordonnées
 - Boîtes englobantes 3D
 - Divers (défini par l'utilisateur)



Les filtres

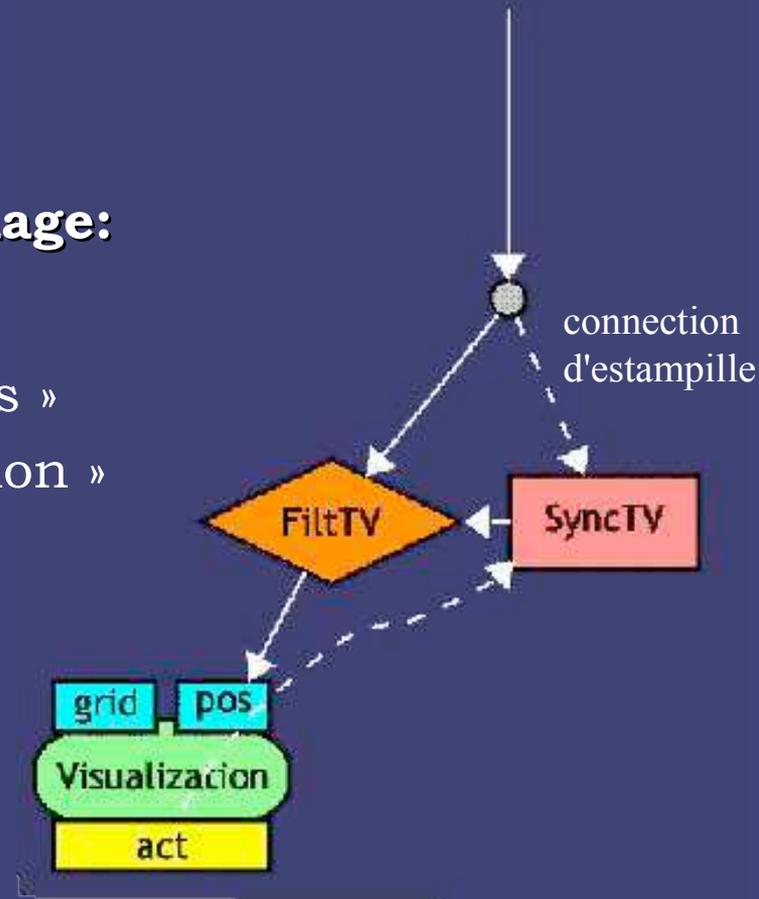


- Toutes ces connections sont FIFO
- Tous les modules sont synchrones

Les Synchroniseurs

- **Implémentent les politiques de couplage:**

- Ils récupèrent les informations
 - « de nouveaux messages sont émis »
 - « des modules ont finis leur itération »
- Ils prennent une décision:
 - filtrage de données et émission
 - routage
 - activation de module
- Ils émettent leurs ordres



- **Visualisation et Tracker devient asynchrones**

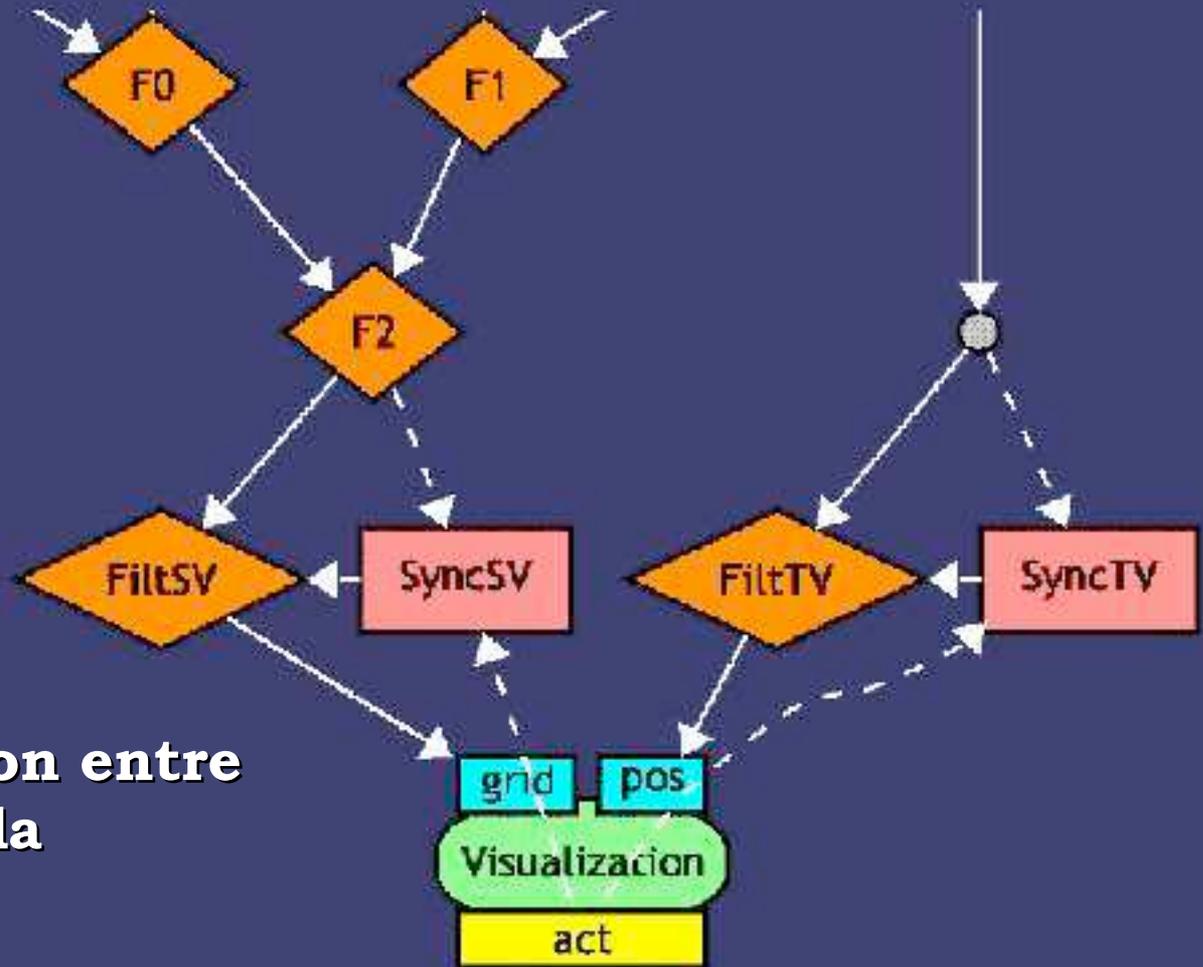
- **Importance de la position du filtre :**

- Charge réseau faible mais latence augmentée
- Charge réseau importante mais latence faible

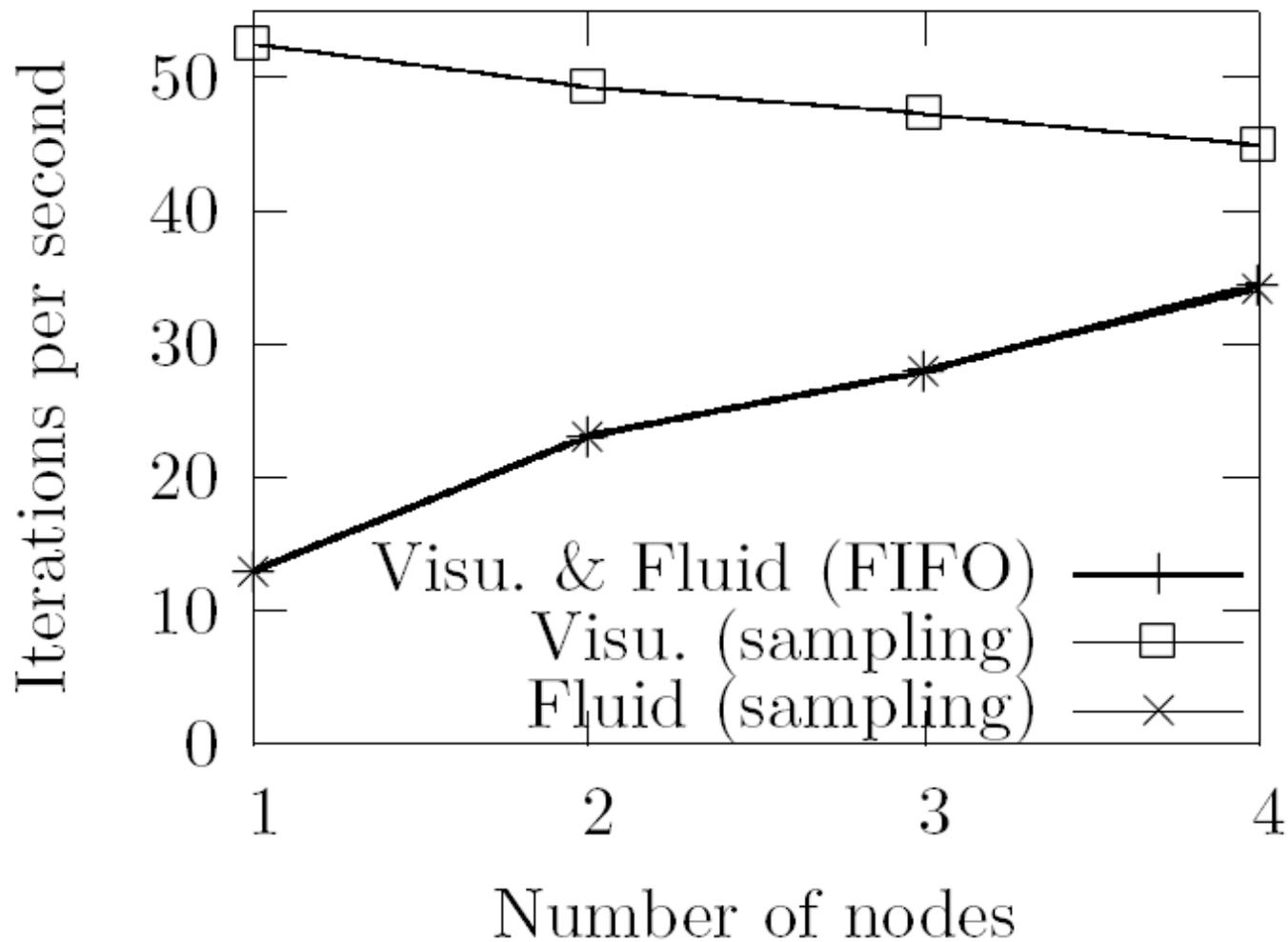
Couplage par contrainte

- **Spécification des contraintes à respecter**
 - Politique des connections
 - FIFO
 - *Glouton*
 - Fréquence max
 - Cohérences des données entre les ports
 - Contraintes spécifique a l'application
 - Zone d'intérêt
 - Bounding box
 - ...
- **Grain de synchronisation variable**
 - Synchroniseur simple implémentant une contrainte bien précise
 - Synchroniseur plus complexe centralisant les contraintes
 - Moteur de résolution de contraintes

Les synchroniseurs

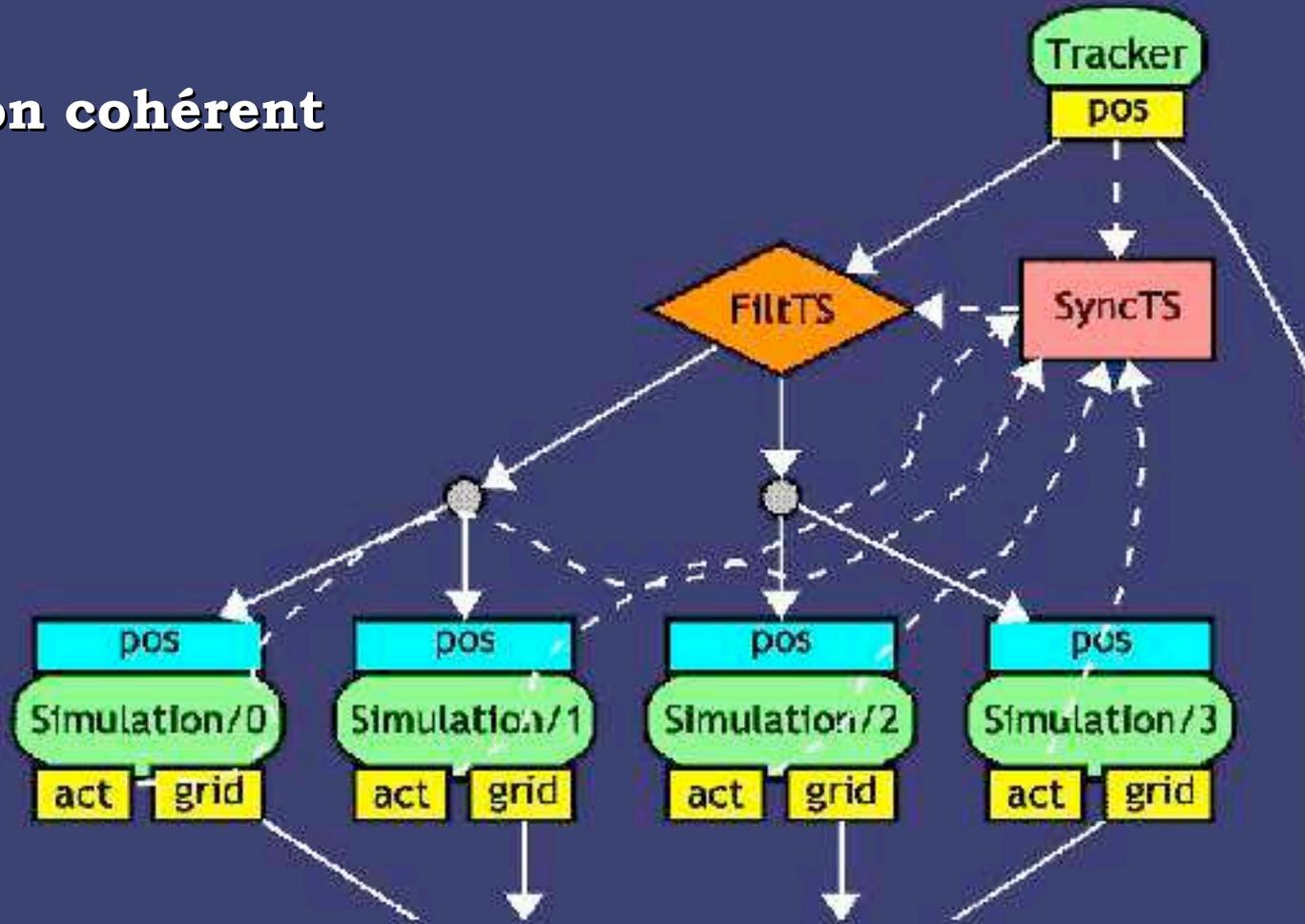


- **Désynchronisation entre la simulation et la visualisation**



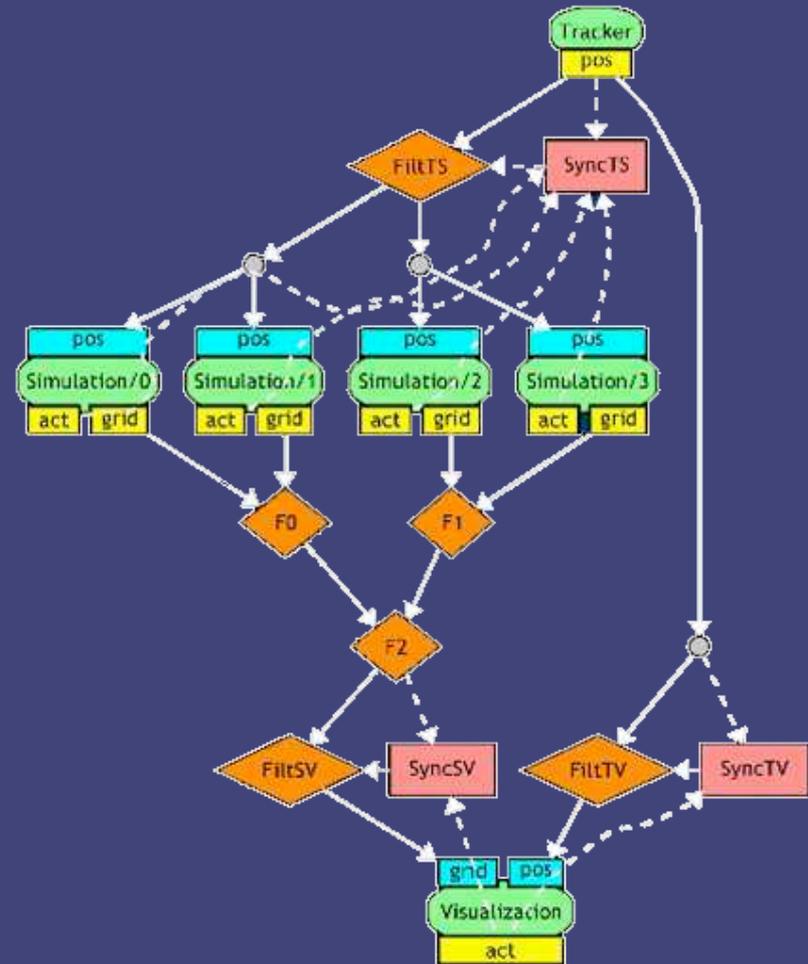
Les synchroniseurs

- Glouton cohérent



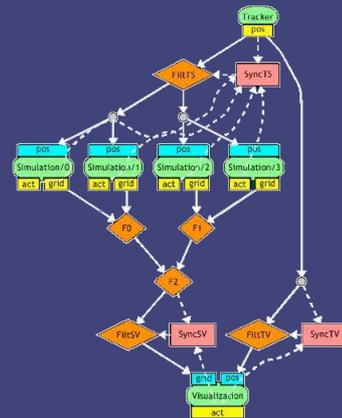
L'application complète

- La prochaine étape : placer ce réseau virtuel sur la machine cible



XML + Parametres

Script (Perl)



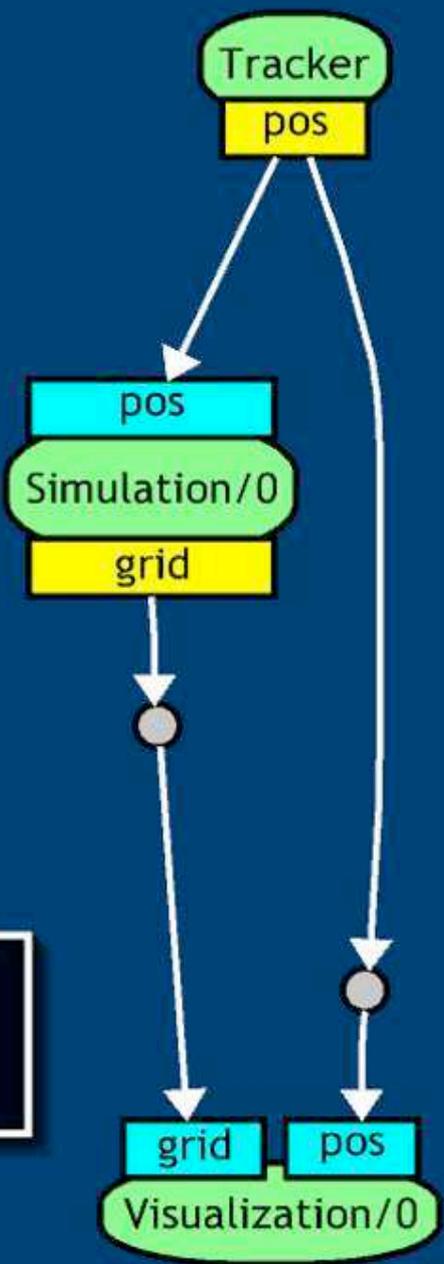
Lancement des modules

Lancement du reseau

L'application fonctionne !!

Implémentation

- **Basé sur un démon tournant sur chaque machine centralisant les communications**
- **Les modules peuvent être dans des processus différents**
- **Un module spécial appelé *contrôleur* est chargé de gérer l'application**
 - Lancement des modules
 - Transmission des commandes bas niveau pour créer les connexions, filtres, synchroniseurs
 - Peut présenter à l'utilisateur une représentation de haut niveau

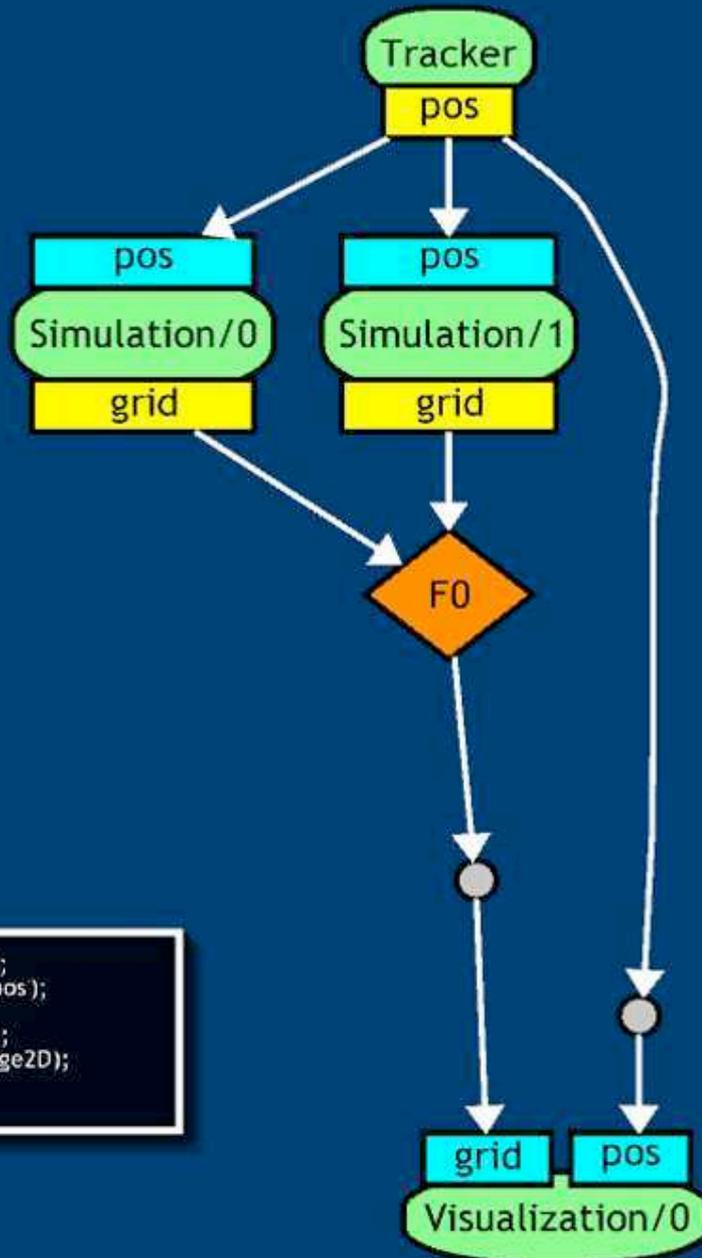


Tracker: 1
 Simulation: 1
 Visualization: 1

```

&addBroadcast('Tracker','pos','Simulation','pos');
&addConnection('Tracker','pos','Visualization','pos');

&addRoutingNodes(R,&getHosts('Simulation'));
&addRecursiveGather('Simulation','grid',R,Merge2D);
&addConnection('R','', 'Visualization','grid');
  
```

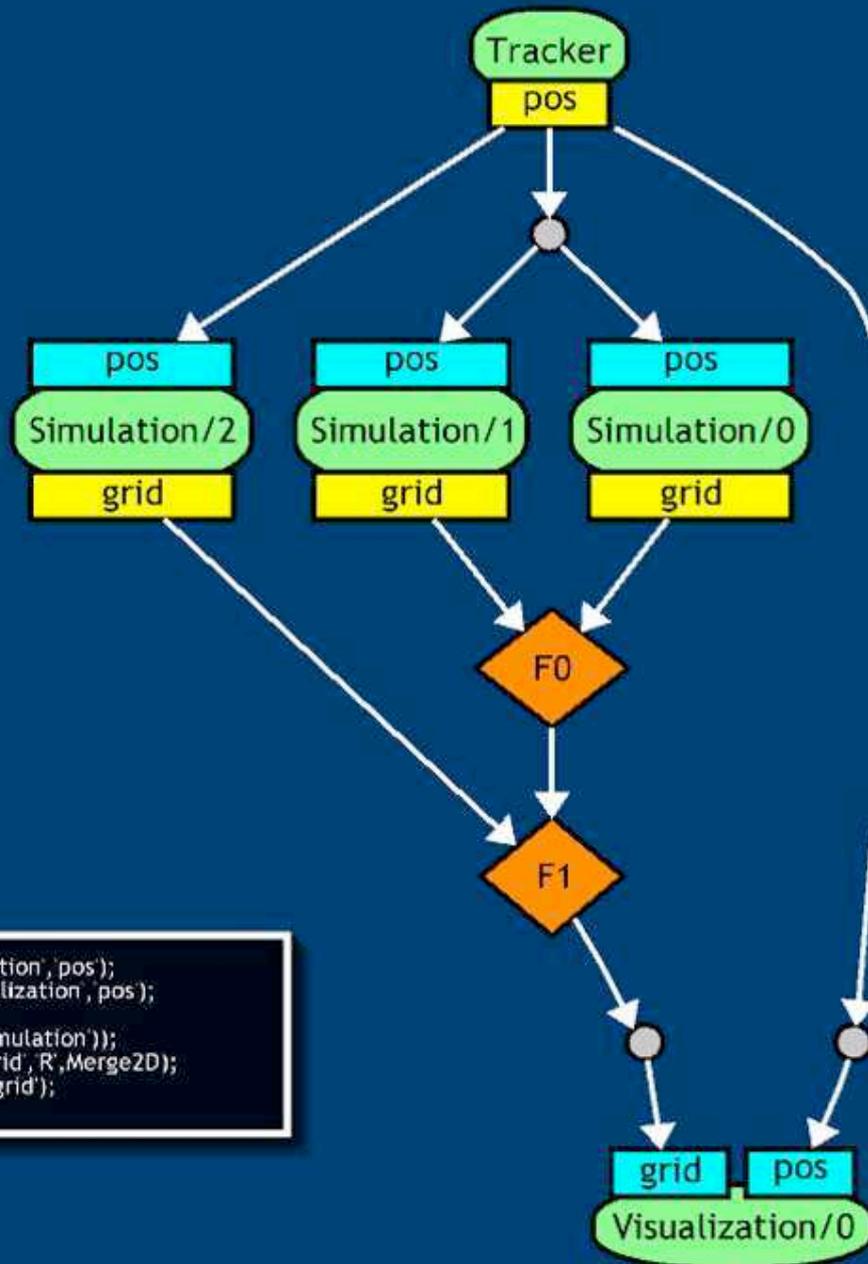


Tracker: 1
 Simulation: 2
 Visualization: 1

```

&addBroadcast('Tracker','pos','Simulation','pos');
&addConnection('Tracker','pos','Visualization','pos');

&addRoutingNodes('R',&getHosts('Simulation'));
&addRecursiveGather('Simulation','grid','R',Merge2D);
&addConnection('R','', 'Visualization','grid');
  
```

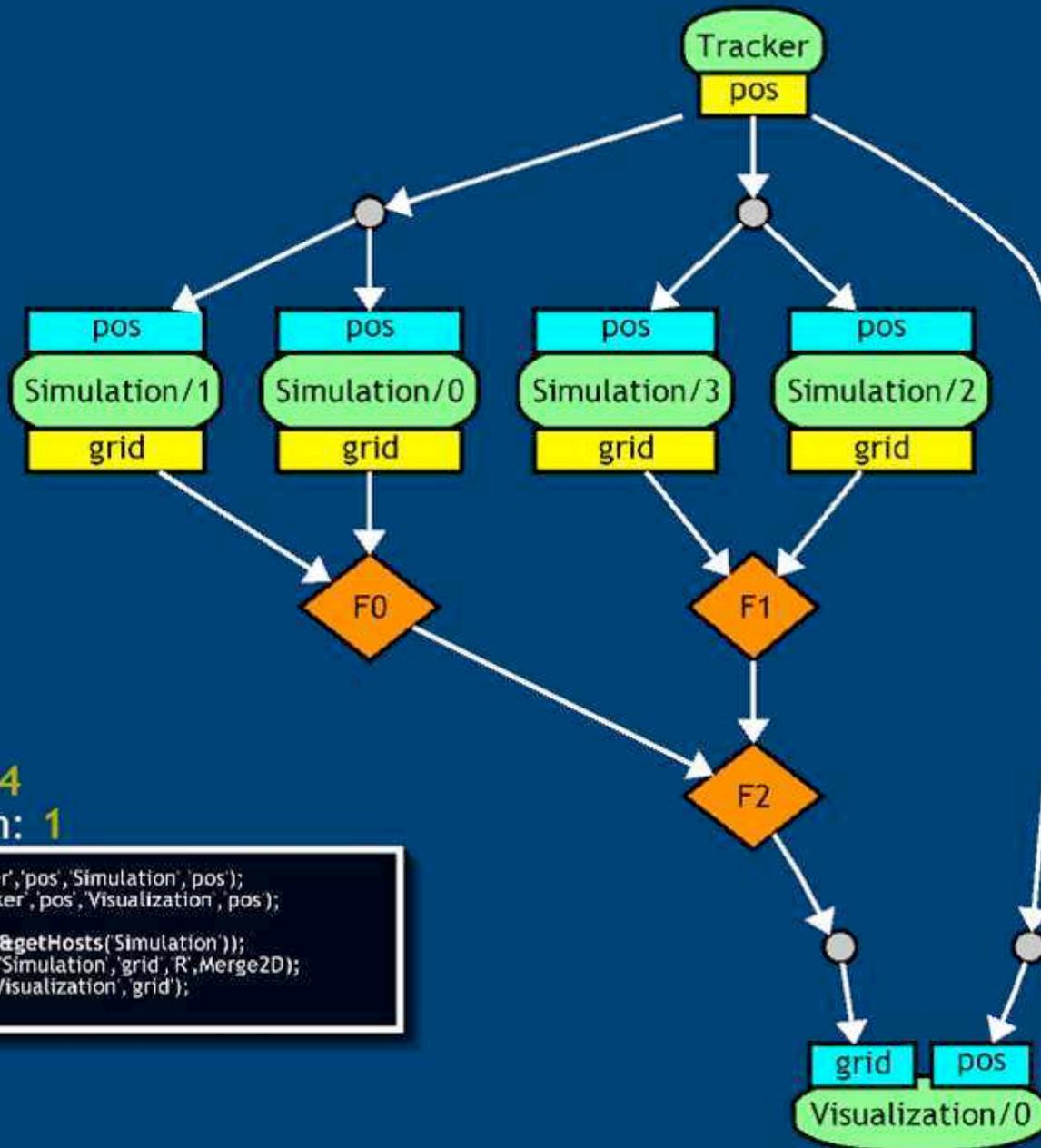


Tracker: 1
 Simulation: 3
 Visualization: 1

```

&addBroadcast('Tracker','pos','Simulation','pos');
&addConnection('Tracker','pos','Visualization','pos');

&addRoutingNodes(R,&getHosts('Simulation'));
&addRecursiveGather('Simulation','grid',R,Merge2D);
&addConnection('R','', 'Visualization','grid');
  
```

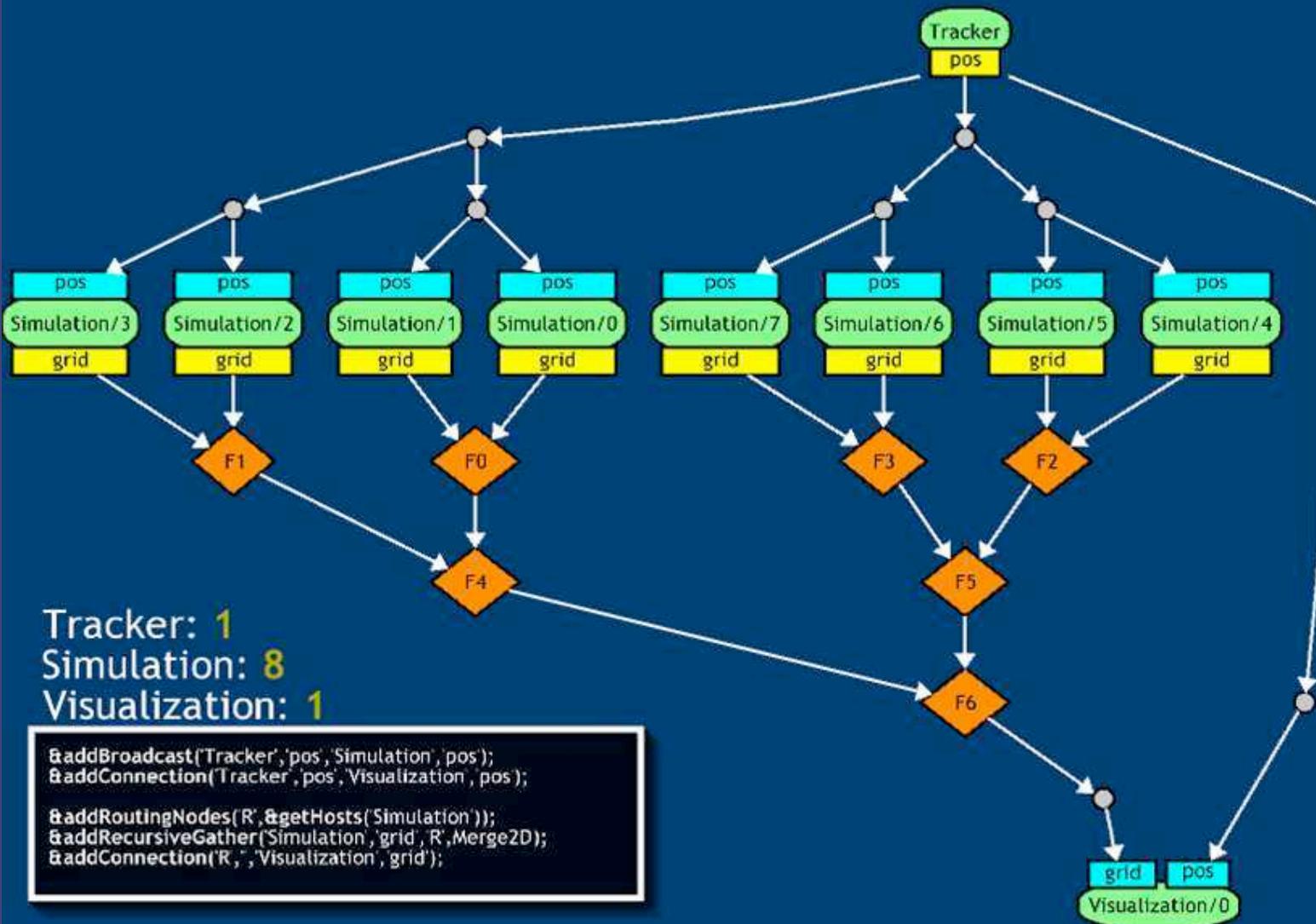


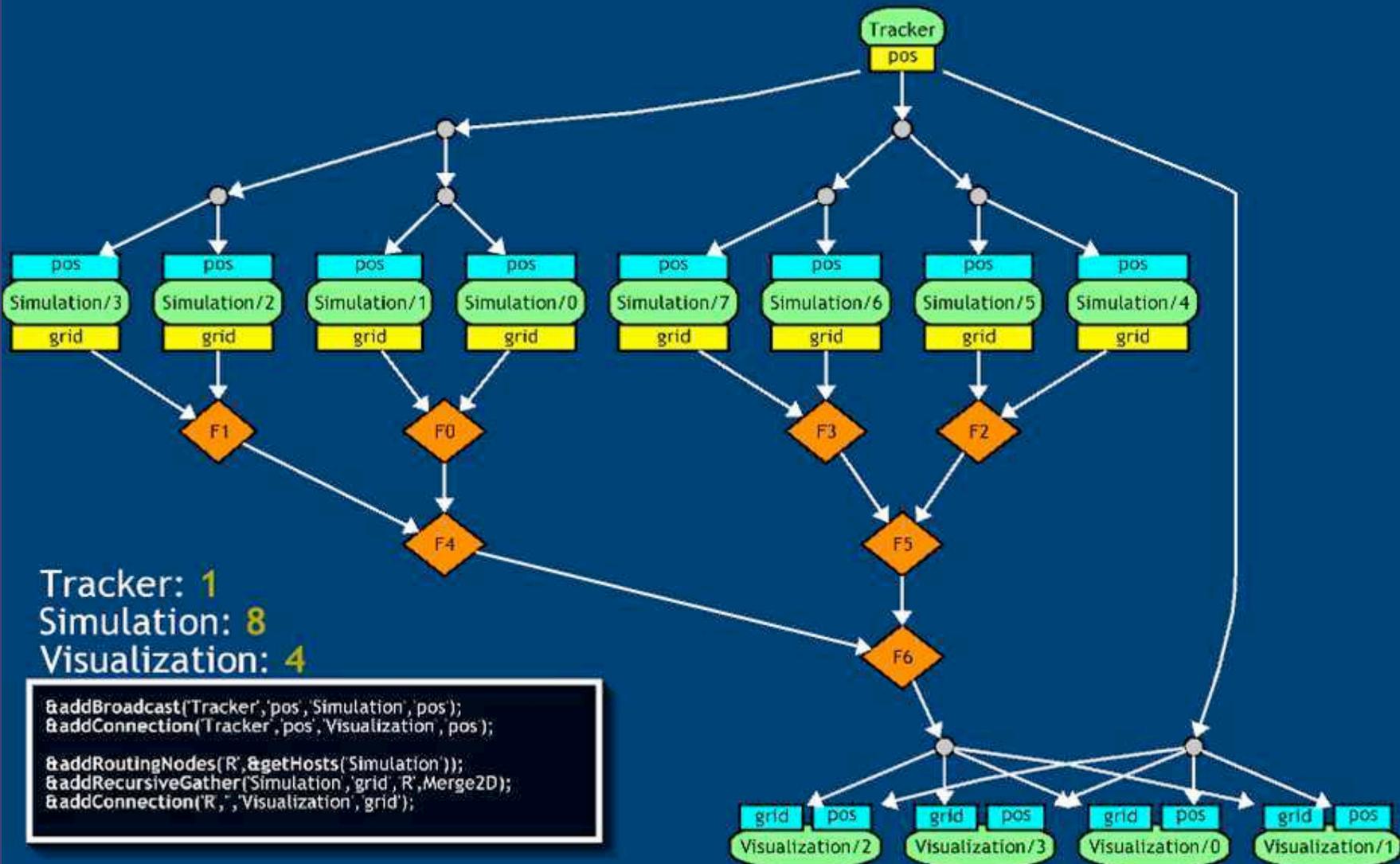
Tracker: 1
 Simulation: 4
 Visualization: 1

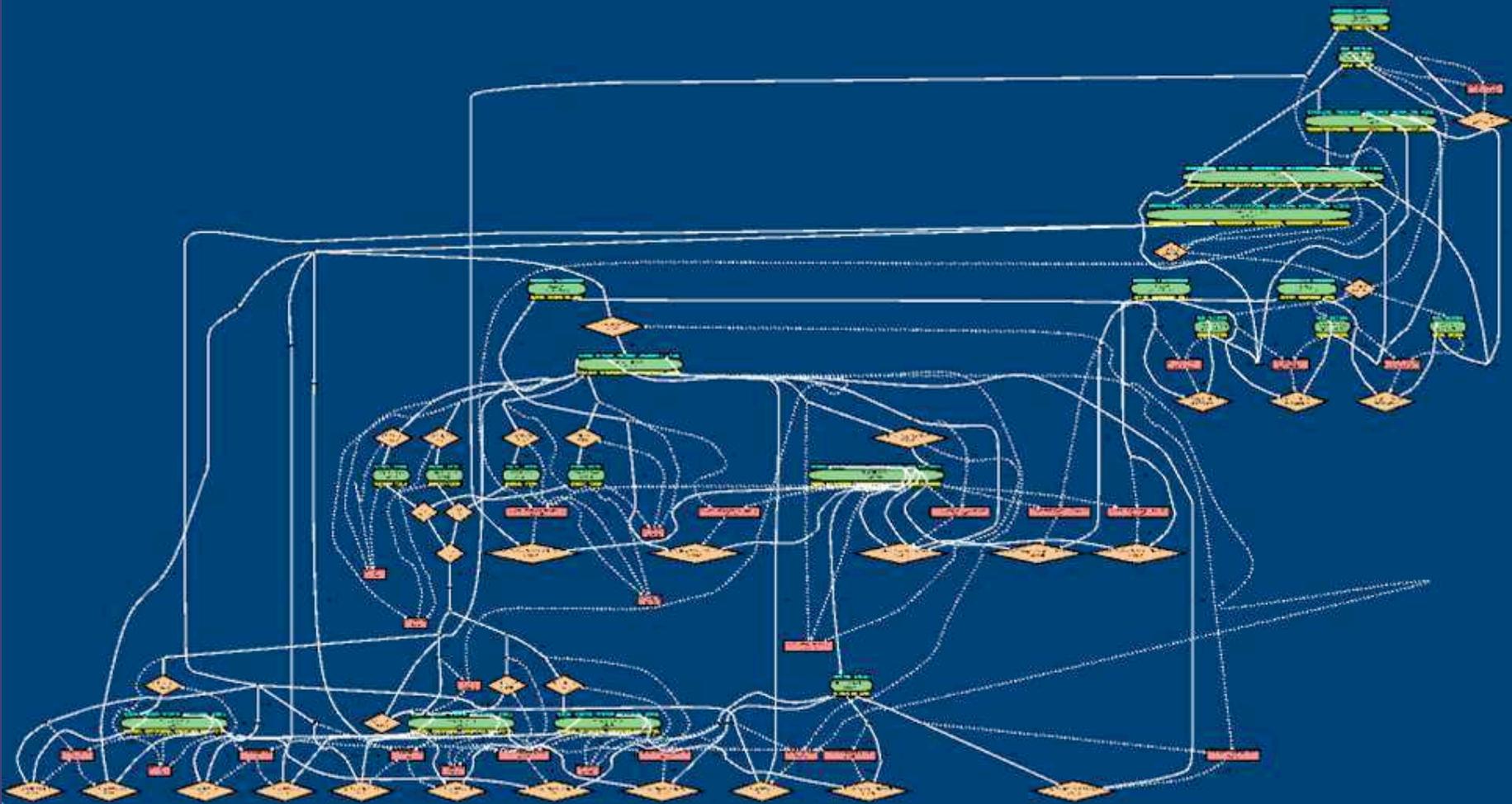
```

&addBroadcast('Tracker','pos','Simulation','pos');
&addConnection('Tracker','pos','Visualization','pos');

&addRoutingNodes(R,&getHosts('Simulation'));
&addRecursiveGather('Simulation','grid',R,Merge2D);
&addConnection('R','', 'Visualization','grid');
  
```







Résultats

- **Meilleures performances qu'avec MPI ou même des communications ad-hoc TCP/UDP**
- **FlowVR est léger et efficace**
- **L'abstraction du réseau FlowVR aide à se concentrer sur les détails pertinents de la création d'une application découplée**
- **Le mécanisme de synchronisation est ad-hoc et non imposé par la librairie utilisée.**
- **Il n'est plus nécessaire de gérer les communications MPI**

Projets

- **Dépliage d'applications FlowVR**
- **Dynamicité des applications FlowVR**
- **Amira et / ou VTK dans FlowVR**
- **Projet TERRE VIRTUELLE et visualisation de données géologiques en RV: intégration du niveau de détail parallèle**