

ARCHITECTURES MATERIELLES ET LOGICIELLES POUR LA REALITE VIRTUELLE

Emmanuel Melin



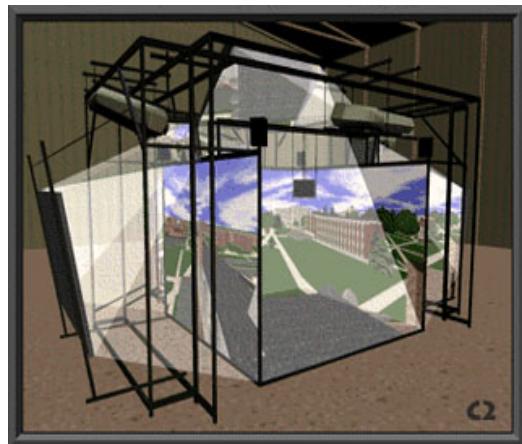
La réalité virtuelle

- L'échange d'informations entre l'homme et la machine est très limité par rapport aux capacités humaines
- En 1965, I. Sutherland et le programme « The ultimate display »
- Accroître l'interaction et la sensation d'immersion de l'utilisateur

Systèmes de RV



la station de calculs

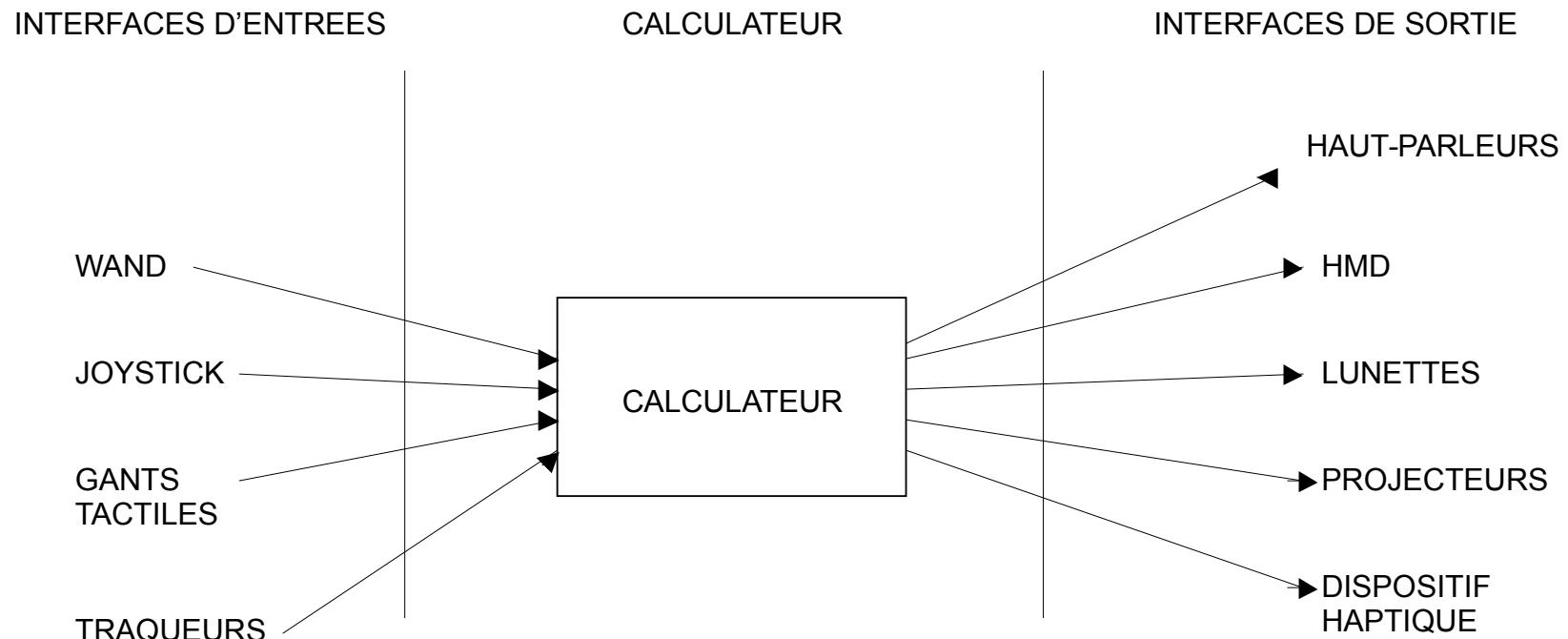


l'affichage



les entrées

Systèmes de réalité virtuelle



La Visualisation

Améliorer la qualité de l'information visuelle :

- Augmenter la taille et la résolution du champ visuel
- Affichage 3D stéréoscopique (vision en relief)
 - > Meilleure sensation d'immersion
 - > Meilleure appréhension des structures 3D

Le Casque (Head Mounted Display)



L'utilisateur porte les écrans près de ses yeux

Limitations technologiques :

- Faible résolution
- Souvent lourd
- Difficile de traquer avec précision les mouvements de la tête :
 - Immersion peu réaliste
 - Malaises
- Utilisateur visuellement isolé:

Le Mur d'images (Reality Center)



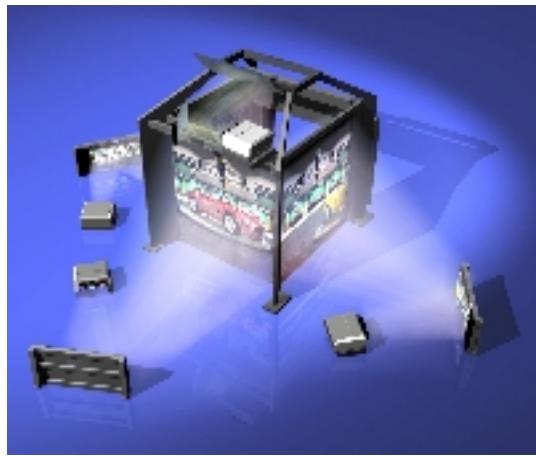
- Plusieurs vidéo-projecteurs sont associés pour former un mur de projection:

- Haute Résolution
- Collaboration
- Travail à l'échelle 1

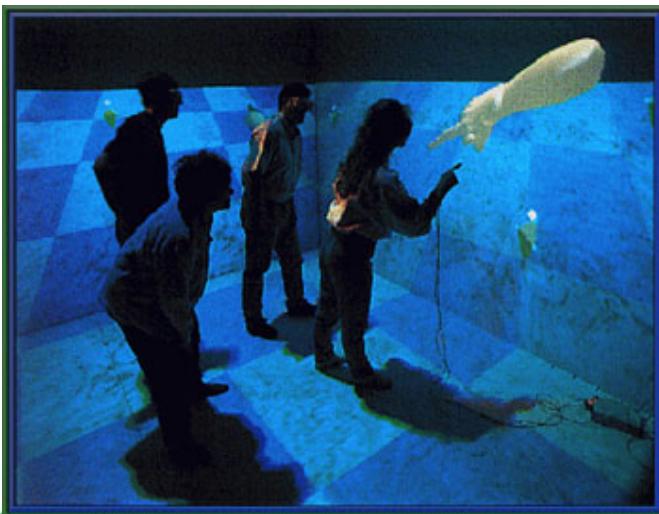
Mais

- Couverture incomplète du champ visuel
- Coût élevé

Le Visio Cube (Cave)



- SigGraph'92 : Premier Cave (Cruz-Neira)
 - > Projections sur 4 faces
- Immersion visuelle complète
 - > Avec la stéréoscopie l'utilisateur ne voit plus la structure cubique
- Travail à l'échelle 1
- Collaboration (limitée en stéréoscopie)
- Mais coût et encombrement très importants



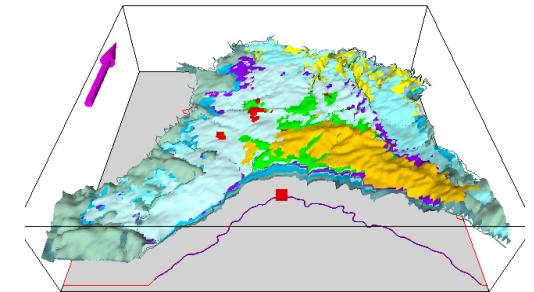
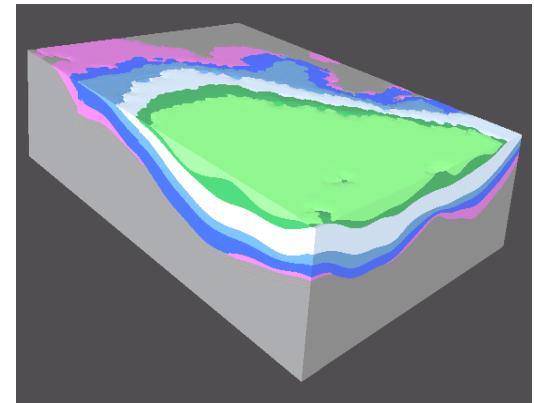
Le Visio Plan (Workbench)



- 93: Premier Workbench créé au GMD (Allemagne)
- Au lieu d'être immergé dans le monde virtuel, l'utilisateur le surplombe
- Couverture limitée du champ visuel
- Collaboration (limitée en stéréoscopie)
- Encombrement et coût raisonnable

-> Opération chirurgicale virtuelle

Le Visio Plan (Workbench)

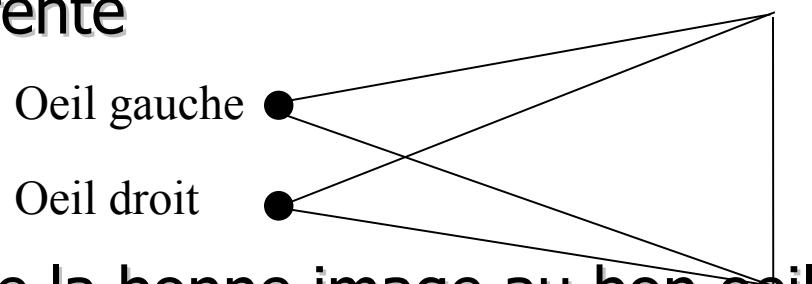


**Visite de la ville de Bruxelles sur le
Workbench du BRGM**

LA STEREOSCOPIE

La Stéréoscopie

- Principe : Chaque oeil doit voir une image avec une perspective légèrement différente



Difficulté: comment transmettre la bonne image au bon oeil

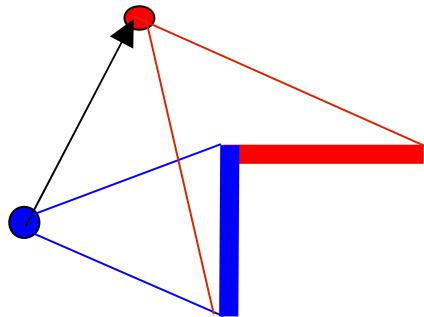
Avec le casque : Un écran pour chaque oeil

Avec le mur d'images, le Cave ou le plan de travail:

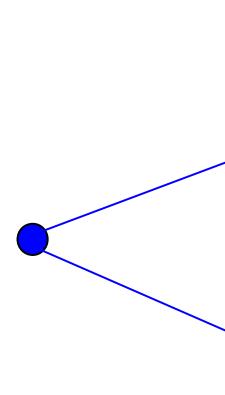
- Stéréoscopie passive
- Stéréoscopie active
- Infitec

Stéréoscopie et Déplacement

Déplacement



Objet 3D réel

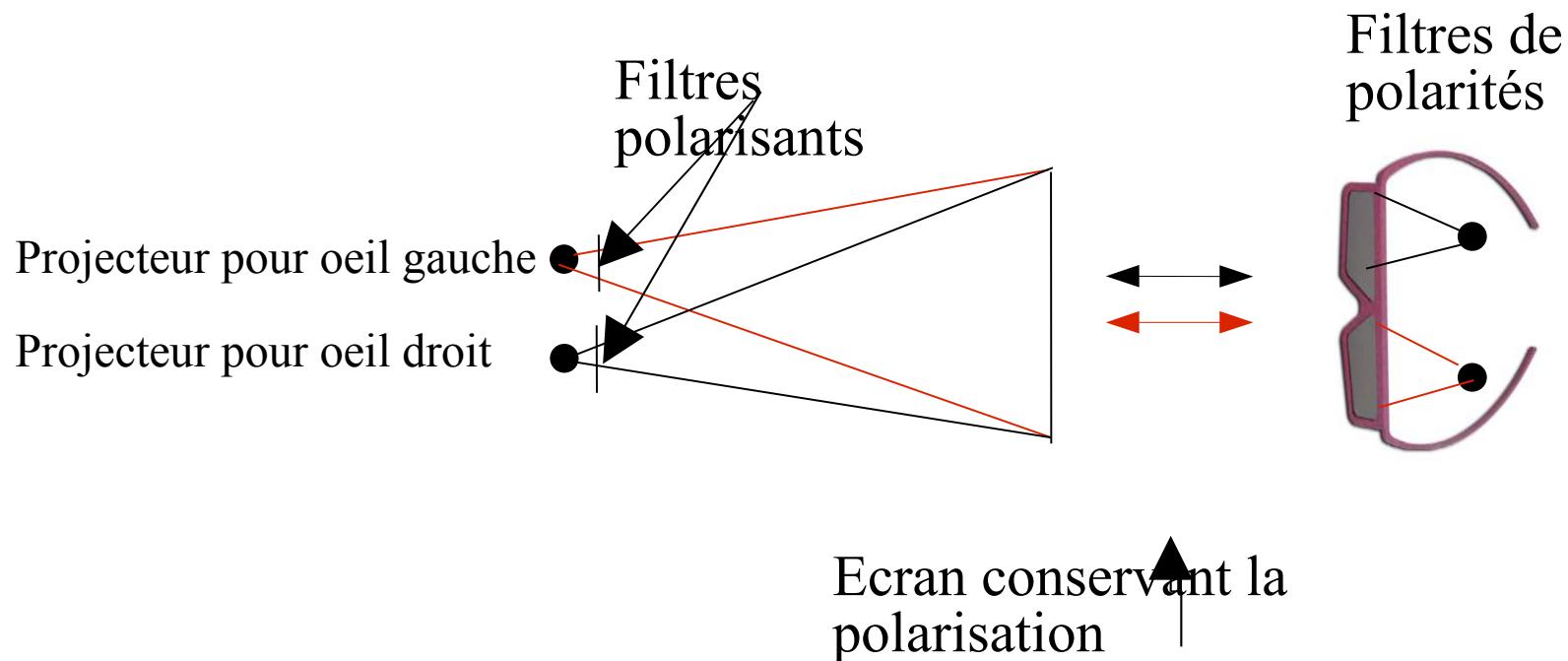


Visualisation stéréoscopique

- La stéréoscopie conduit naturellement à vouloir se déplacer autour des volumes
- Pour ne pas perdre l'effet stéréoscopique, la position de l'utilisateur est traquée et les images gauches et droites recalculées en fonction de cette position (en temps-réel)

La Stéréoscopie Passive

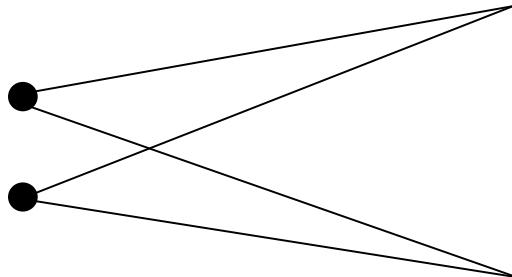
- Principe: images séparées par polarisation des sources lumineuses



- Technique habituellement utilisée dans les parcs d'attraction
- La polarisation induit une perte de luminosité
- Lunettes bon marché mais équipement de projection coûteux

Stéréoscopie Active

Oeil gauche



Oeil droit



Image 1

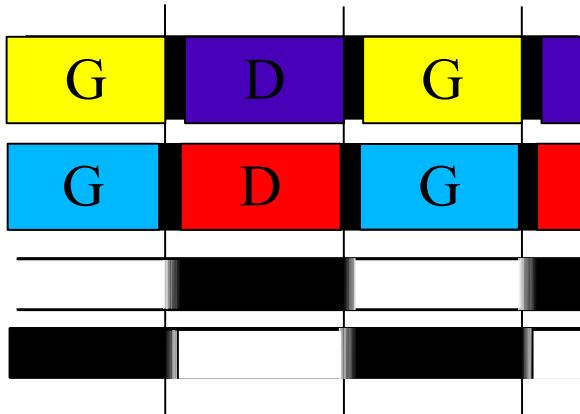


Image 2

Oeil Gauche

Oeil Droit

Le système calcule
2 images

Projection alternée de l'image
gauche et de l'image droite.

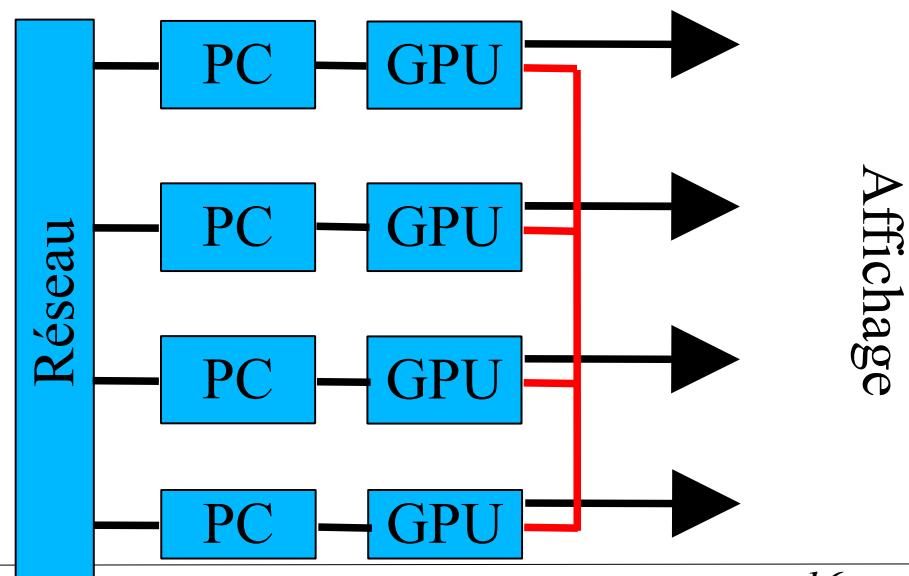
Alternance oeil gauche, oeil
droit avec des lunettes à
obturateurs.

Synchronisation entre les
images et les lunettes à
obturateurs

La Stéréoscopie Active

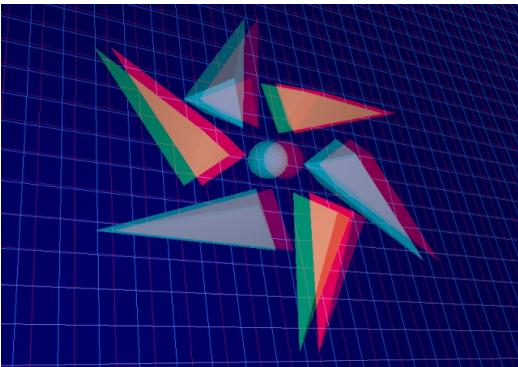


- ❖ Coût élevé des lunettes
- ❖ Divise par deux le taux de rafraîchissement de l'écran
 - 100Hz -> 50Hz pour chaque œil
- ❖ Multi-écrans: synchronisation du balayage (genlock)



Infitec (www.tan.de)

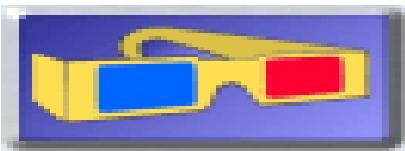
❖ La technique de filtrage par longueurs d'ondes :



❖ L'image gauche est imprimée en bleu, la droite en rouge

❖ Les verres des lunettes bloquent le bleu ou le rouge par l'utilisation de verres colorés

-> dégradation des couleurs



❖ Infitec (Tan pour DaimlerChrysler)

❖ Reprend la technique des couleurs, mais en jouant sur un décalage très fin en longueur d'onde de l'image gauche et de l'image droite

-> Faible perte de luminosité

-> Altération des couleurs quasi imperceptible



Stéréoscopie Traquée à Plusieurs

- Nombre d'images à calculer = $2 * \text{Nb_utilisateurs_traqués}$
- Affichage stéréoscopique :
 - Stéréoscopie passive :
 - $\text{Nb_projecteurs} = 2 * \text{Nb_utilisateurs_traqués} * \text{Nb écrans}$
 - Stéréoscopie active :
 - Taux de rafraîchissement divisé par
 $2 * \text{Nb_utilisateurs_traqués}$
 - Infitec :
 - Décaler dans le spectre des longueurs d'ondes les
 $2 * \text{Nb_utilisateurs_traqués}$ images
- Actuellement on se limite à 1 voir 2 utilisateurs traqués en stéréo active

Systèmes de réalité virtuelle: le calculateur

- Gestion des interfaces d'entrée
 - Gestion des interfaces de sortie
 - Calculs de pré-rendu(simulation,...)
 - Calculs de rendus(image, son,...)
- ☒ **Machine puissante avec éventuellement plusieurs sorties vidéos**

Systèmes de réalité virtuelle: le calculateur

- Un exemple : le SGI ONYX
- Multiprocesseur, multipipe, multicanal
- Machine haute performance mais non extensible, coût élevé (~ 500.000\$)



Los Alamos Koolaid Cluster

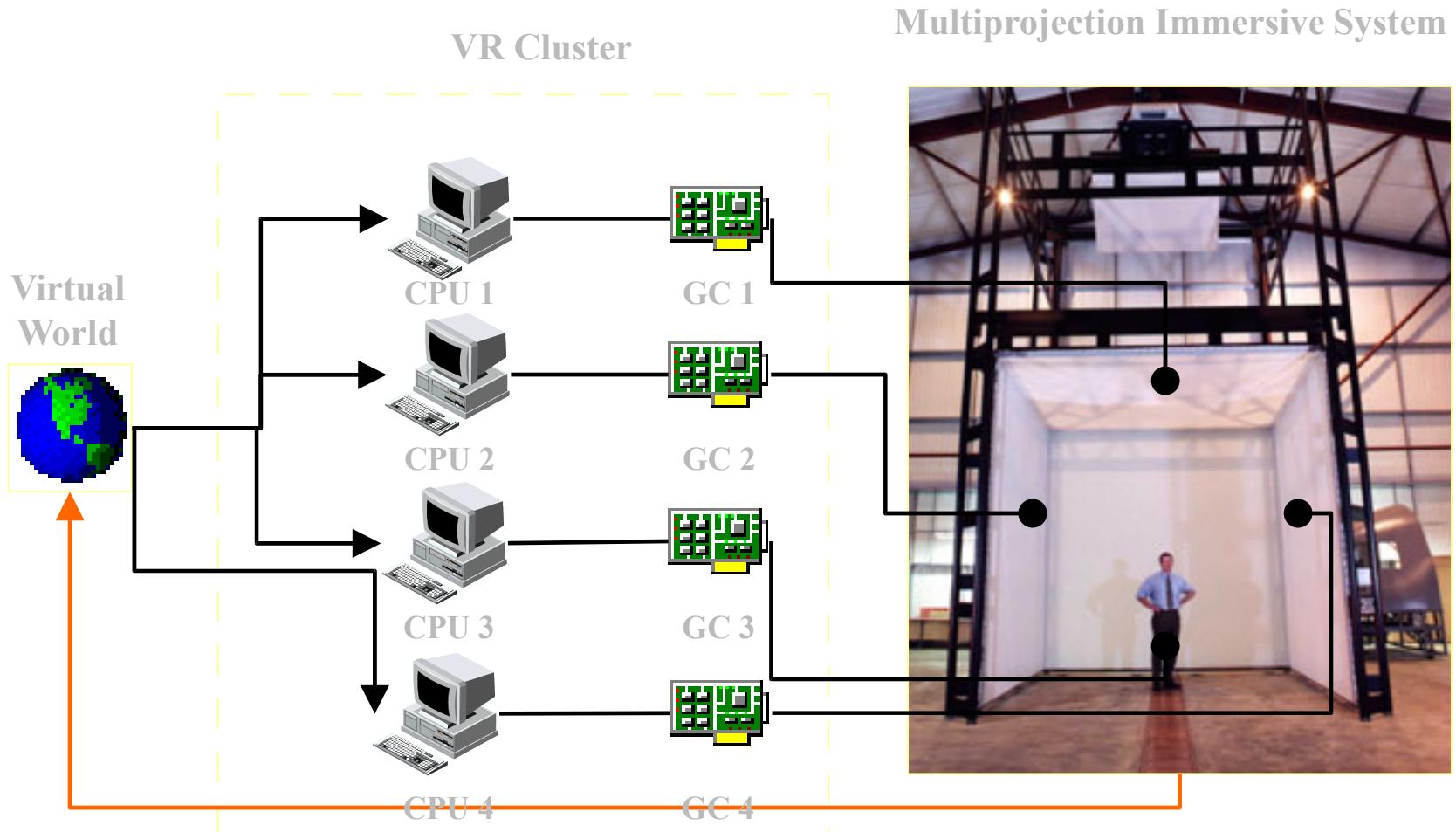


VR CLUSTER

VR Cluster, pourquoi?

- Coût
 - Fabrication à grande échelle
- Flexibilité
 - Diversité d' options d'intégration
- Accès
 - Partout, tout temps, à quiconque
- Evolutivité
- Disponibilité
 - Large gamme de pièces disponibles (cartes graphiques, CPU, ...)

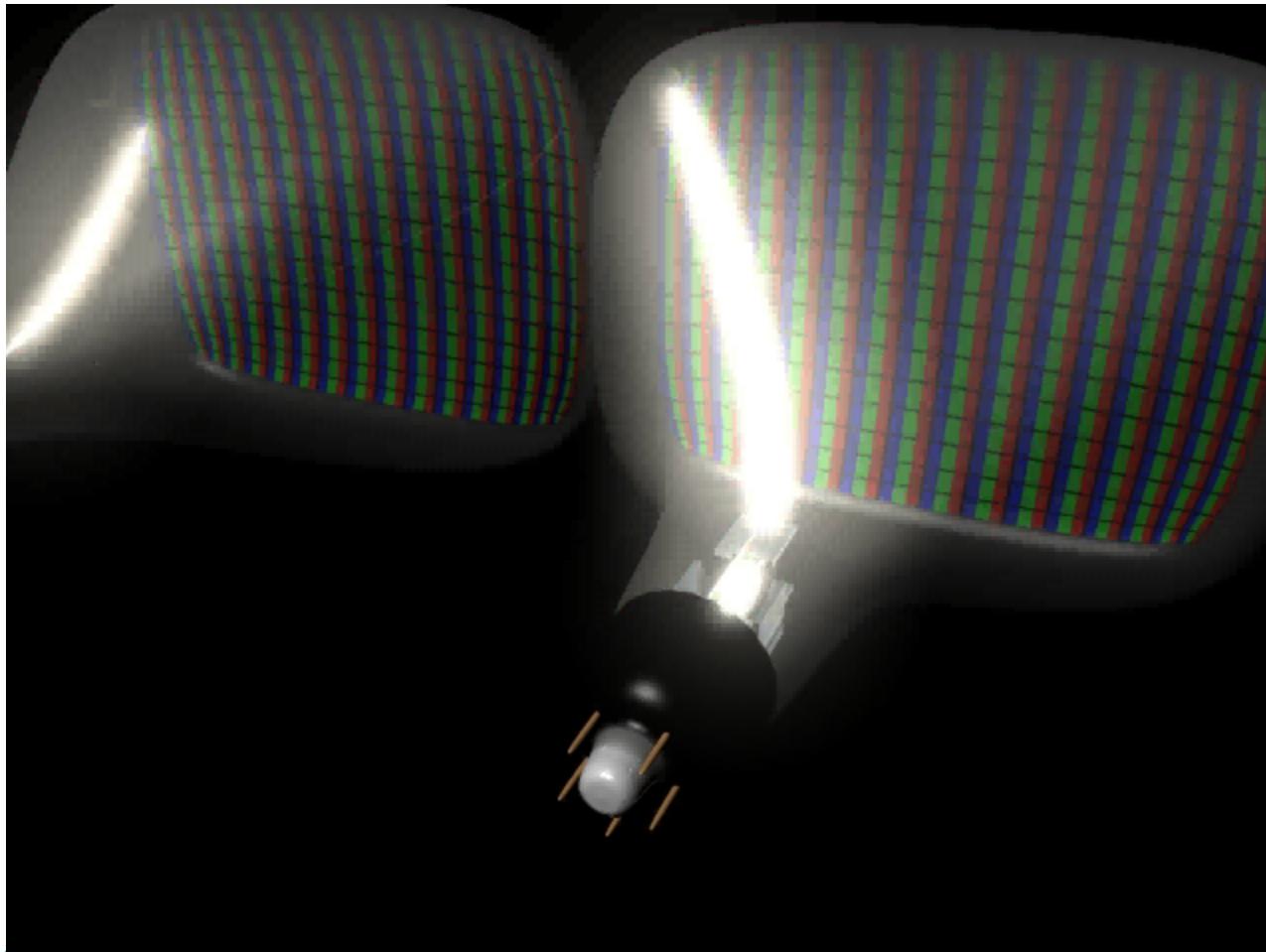
Anatomie du VR cluster



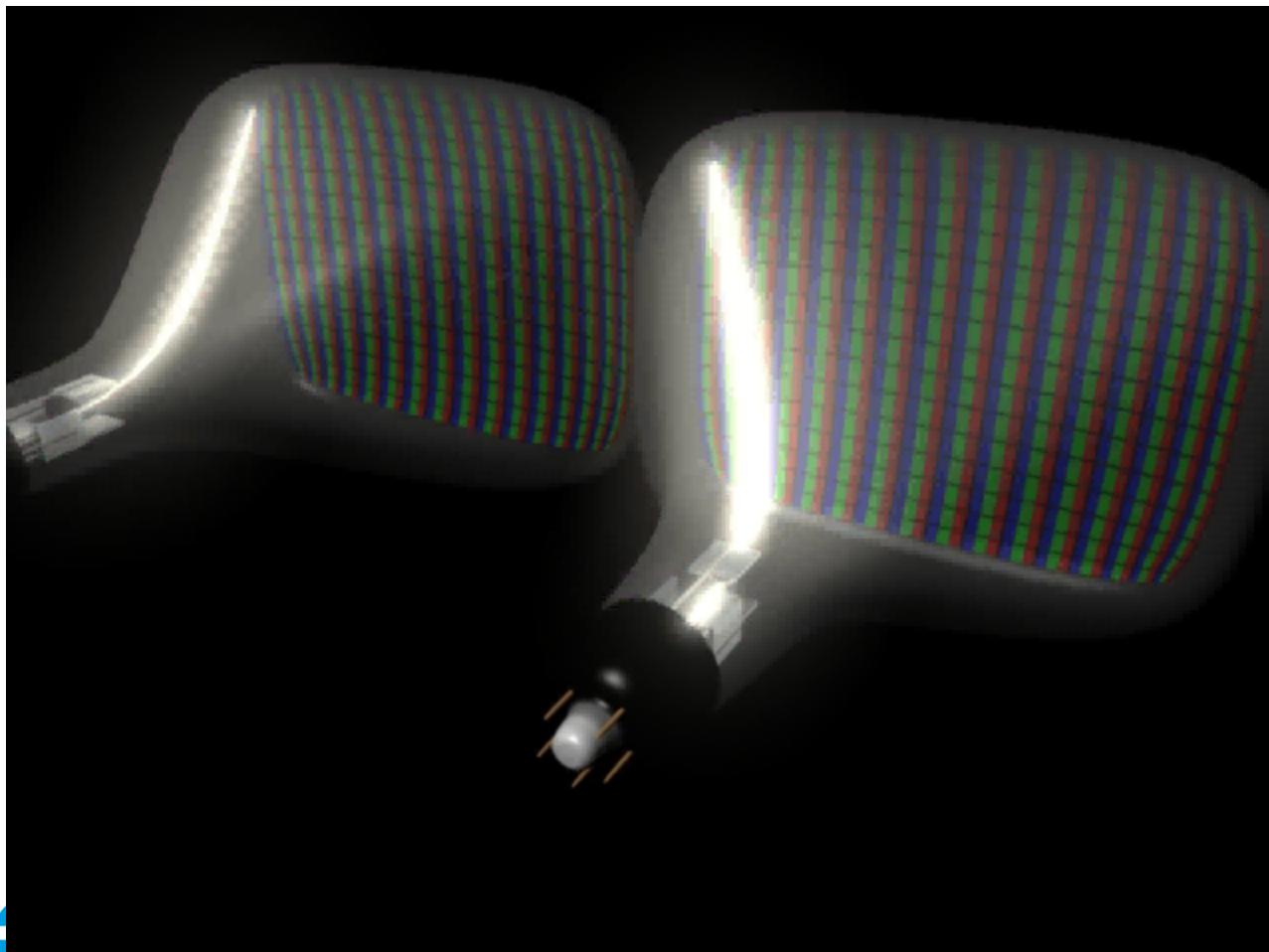
Containtes du VR Cluster

- 3 contraintes à satisfaire:
 - Genlock
 - Swap-lock ou frame-lock
 - Data-lock
- Différentes solutions, matérielles et logicielles, existent

Deux CRT sans Genlock



Deux CRT avec Genlock



SoftGenLock

Approche logicielle pour la stéréo active et le genlock pour les clusters Linux
(netjuggler.sourceforge.net)

→ Marche potentiellement avec toute carte graphique compatible VGA (niveau registres)



SoftGenLock



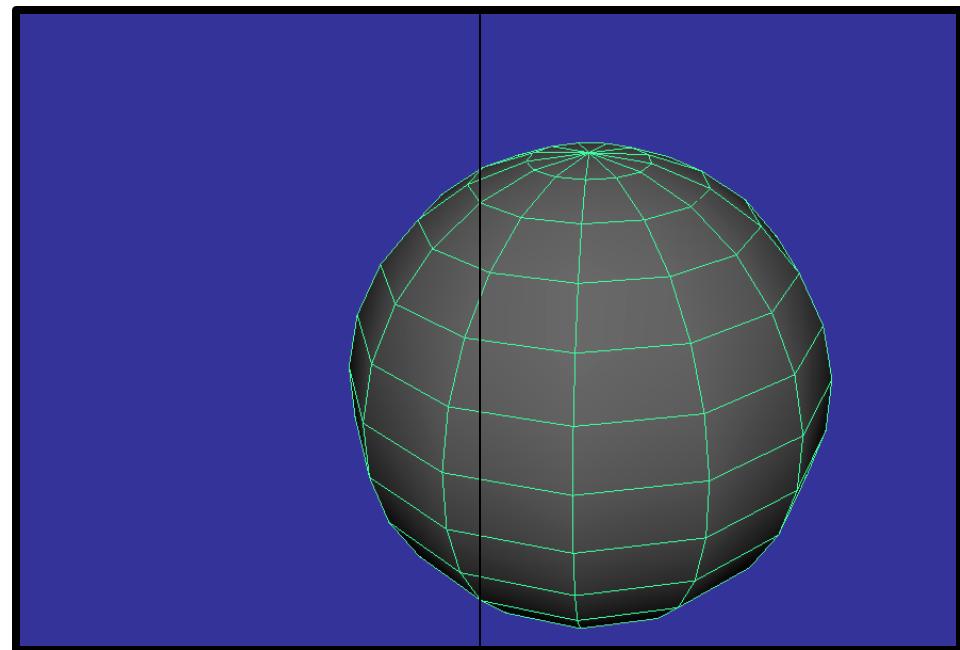
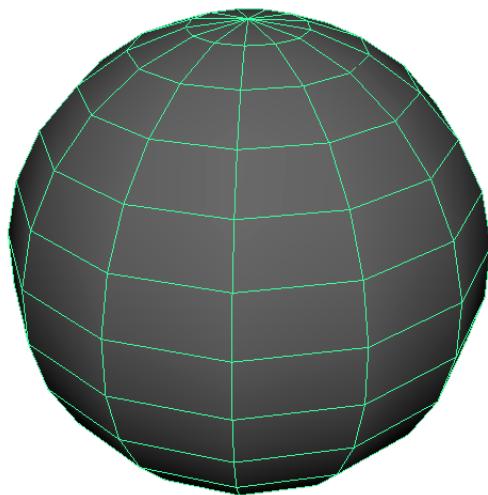
- Envoi du signal de flip aux lunettes
- S'apprêter à afficher l'image droite



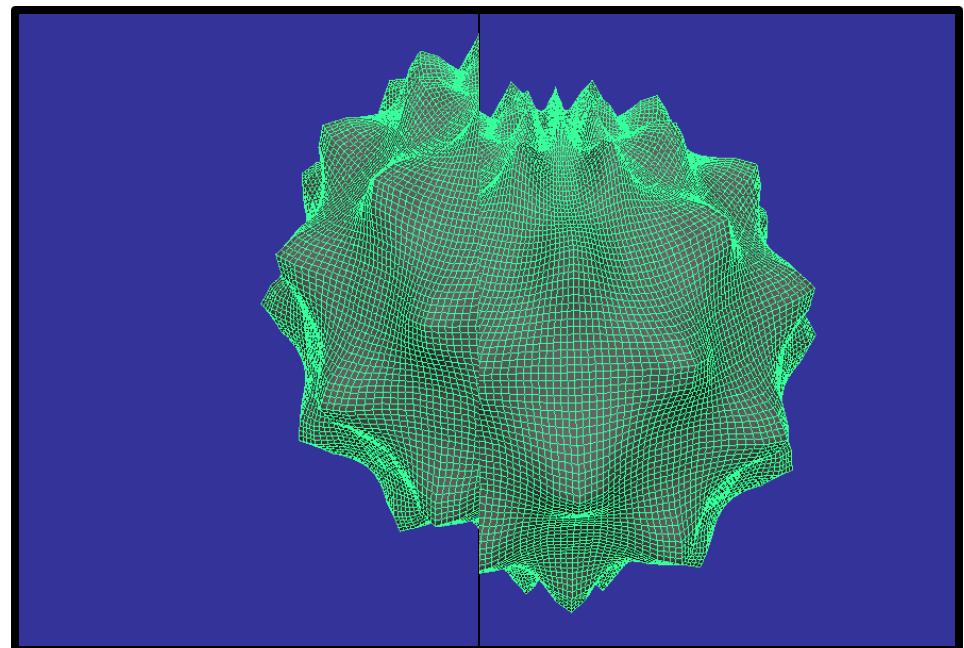
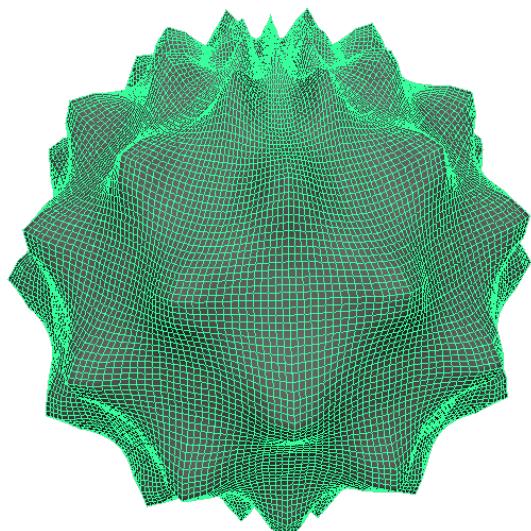
Swap-Lock

- Les différentes vues d'une scène peuvent nécessiter des temps de calcul différents.
- Les swaps des framebuffers doivent être synchronisés.
Sinon, cassures.
- Solution logicielle(réseau de synchronisation) ou matérielle (cartes Wildcat)

Scène avec swap-Lock



Scène sans swap-Lock



Data-Lock

- Chaque nœud dessine ses frames en utilisant de l'information stockée localement
- Pour que les vues soient cohérentes, les informations sur les nœuds doivent être consistantes
- Solution logicielle seulement.

Lifo Black Cluster

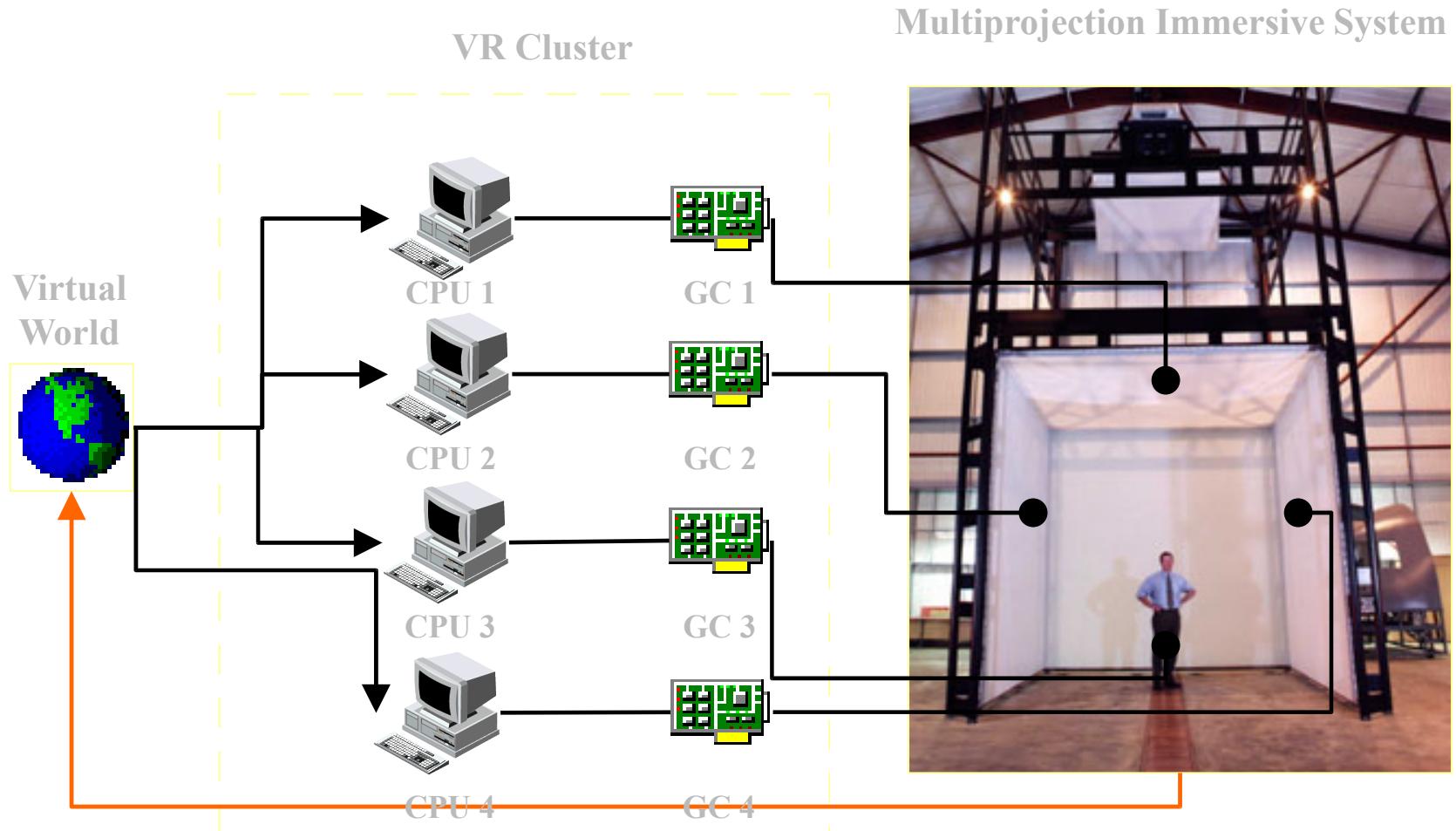


VR CLUSTER

VR Cluster, pourquoi?

- Coût
 - Fabrication à grande échelle
- Flexibilité
 - Diversité d' options d'intégration
- Accès
 - Partout, tout temps, à quiconque
- Evolutivité
- Disponibilité
 - Large gamme de pièces disponibles (cartes graphiques, CPU, ...)

Anatomie du VR cluster



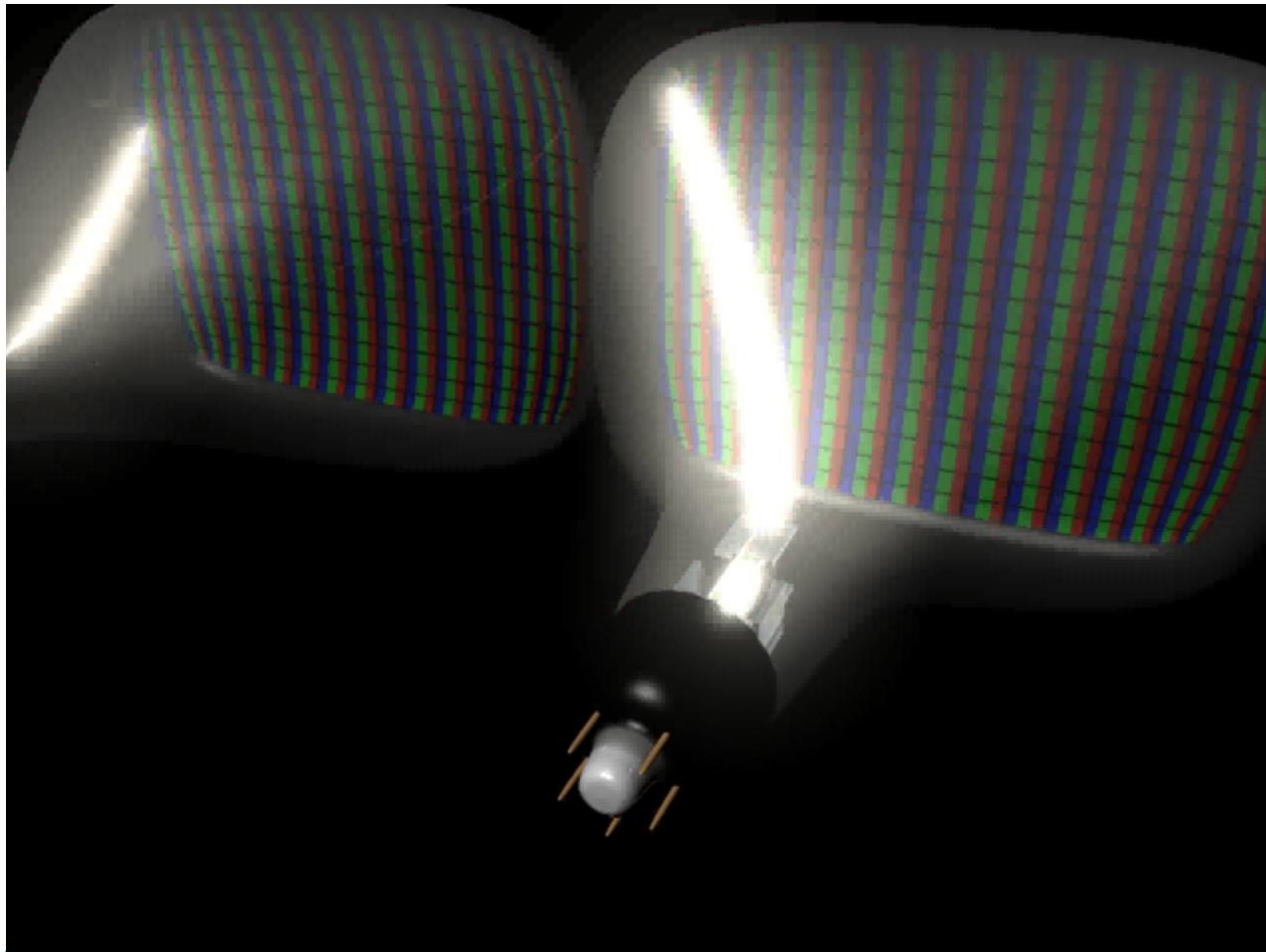
Containtes du VR Cluster

- 3 contraintes à satisfaire:
 - Genlock
 - Swap-lock ou frame-lock
 - Data-lock
- Différentes solutions, matérielles et logicielles, existent

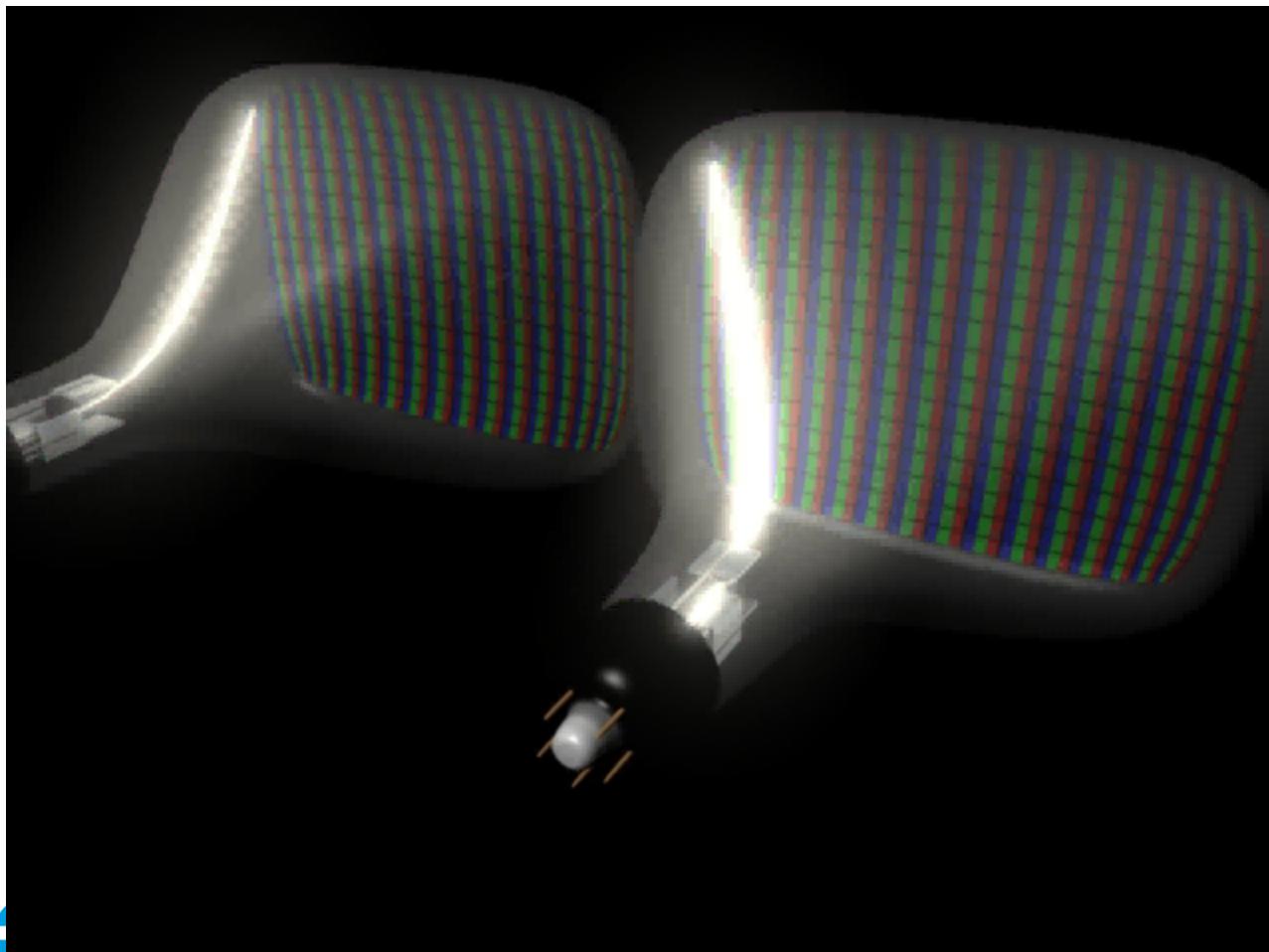
Genlock

- Les nœuds du cluster produisent une séquence de frames vidéo.
- Pour que l'image globale sur tous les écrans du cluster soit cohérente, les productions doivent être synchronisées
- Critique pour la stéréoscopie active .
- Solutions matérielles(Wildcat) et mixtes (SoftGenLock).

Deux CRT sans Genlock



Deux CRT avec Genlock



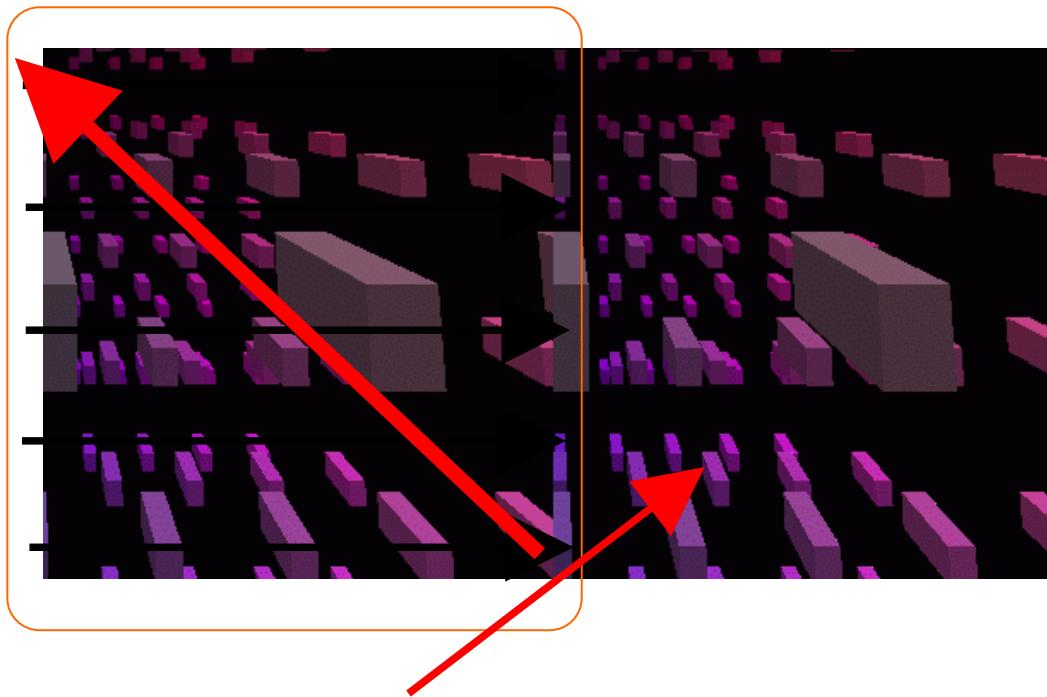
SoftGenLock

Approche logicielle pour la stéréo active et le genlock pour les clusters Linux
(netjuggler.sourceforge.net)

→ **Marche potentiellement avec toute carte graphique compatible VGA (niveau registres)**



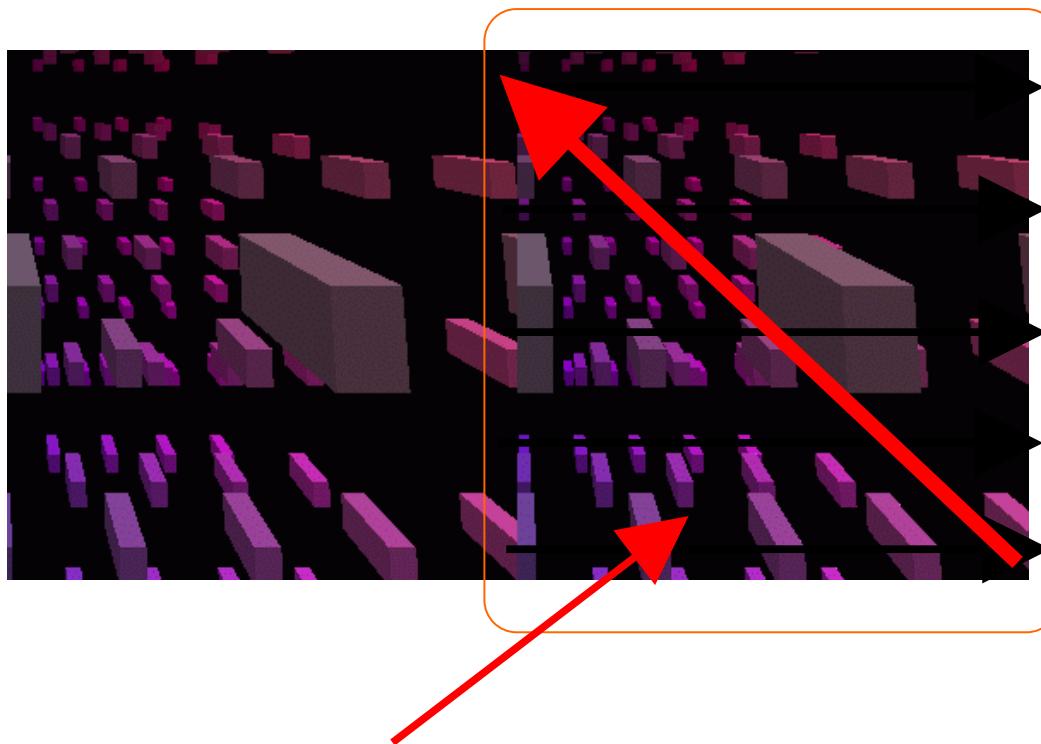
SoftGenLock



- Envoi du signal de flip aux lunettes
- S'apprêter à afficher l'image droite



SoftGenLock



Blanking vidéo:

Envoi du signal de flip aux lunettes

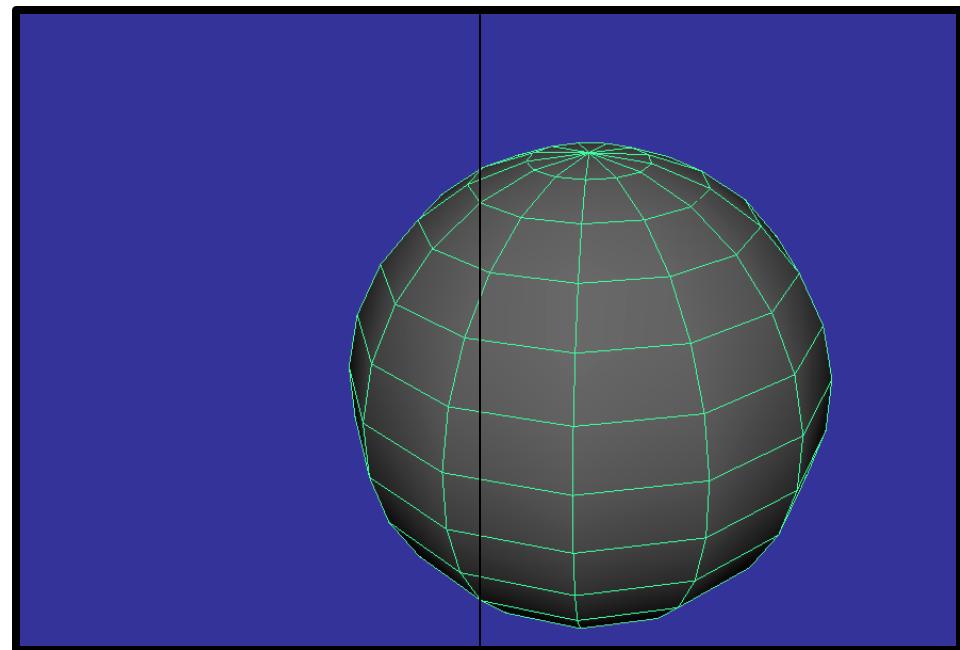
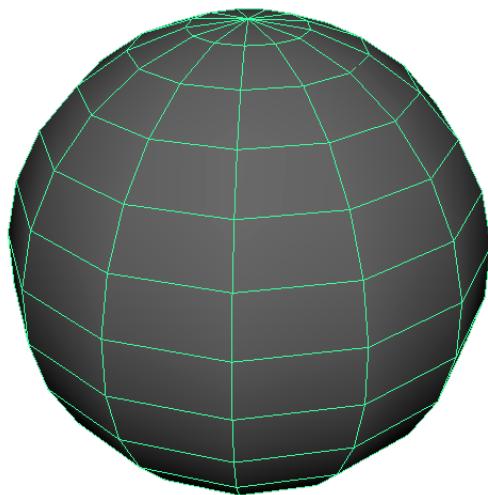
- S'apprêter à afficher l'image gauche



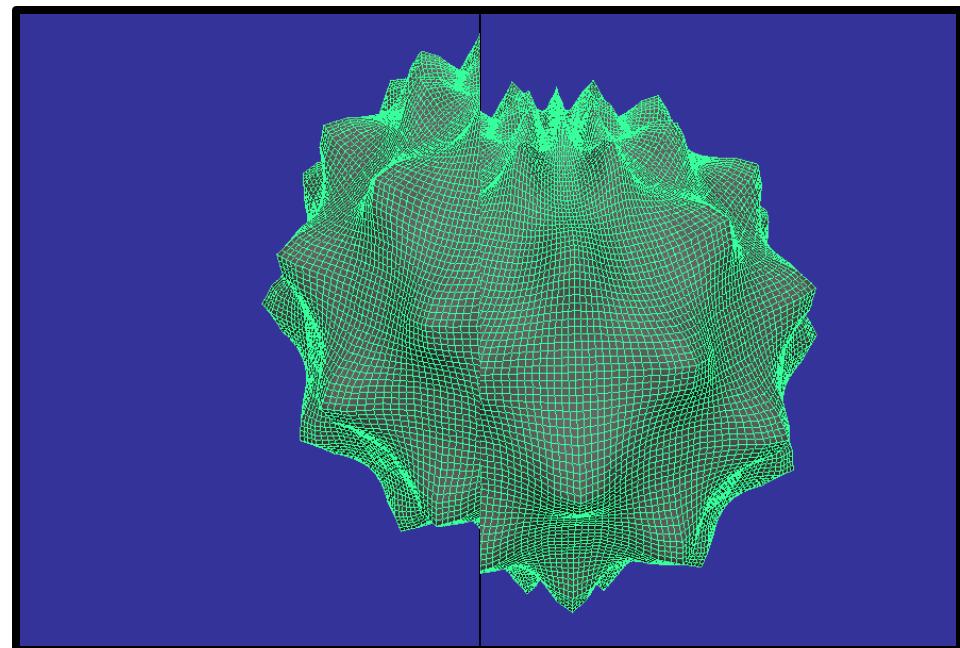
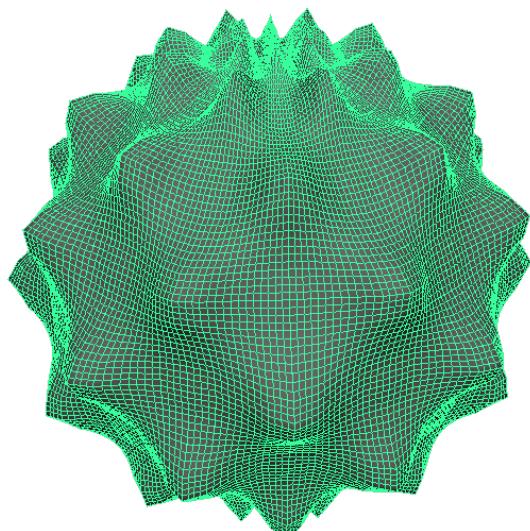
Swap-Lock

- Les différentes vues d'une scène peuvent nécessiter des temps de calcul différents.
- Les swaps des framebuffers doivent être synchronisés.
Sinon, cassures.
- Solution logicielle(réseau de synchronisation) ou matérielle (cartes Wildcat)

Scène avec swap-Lock



Scène sans swap-Lock



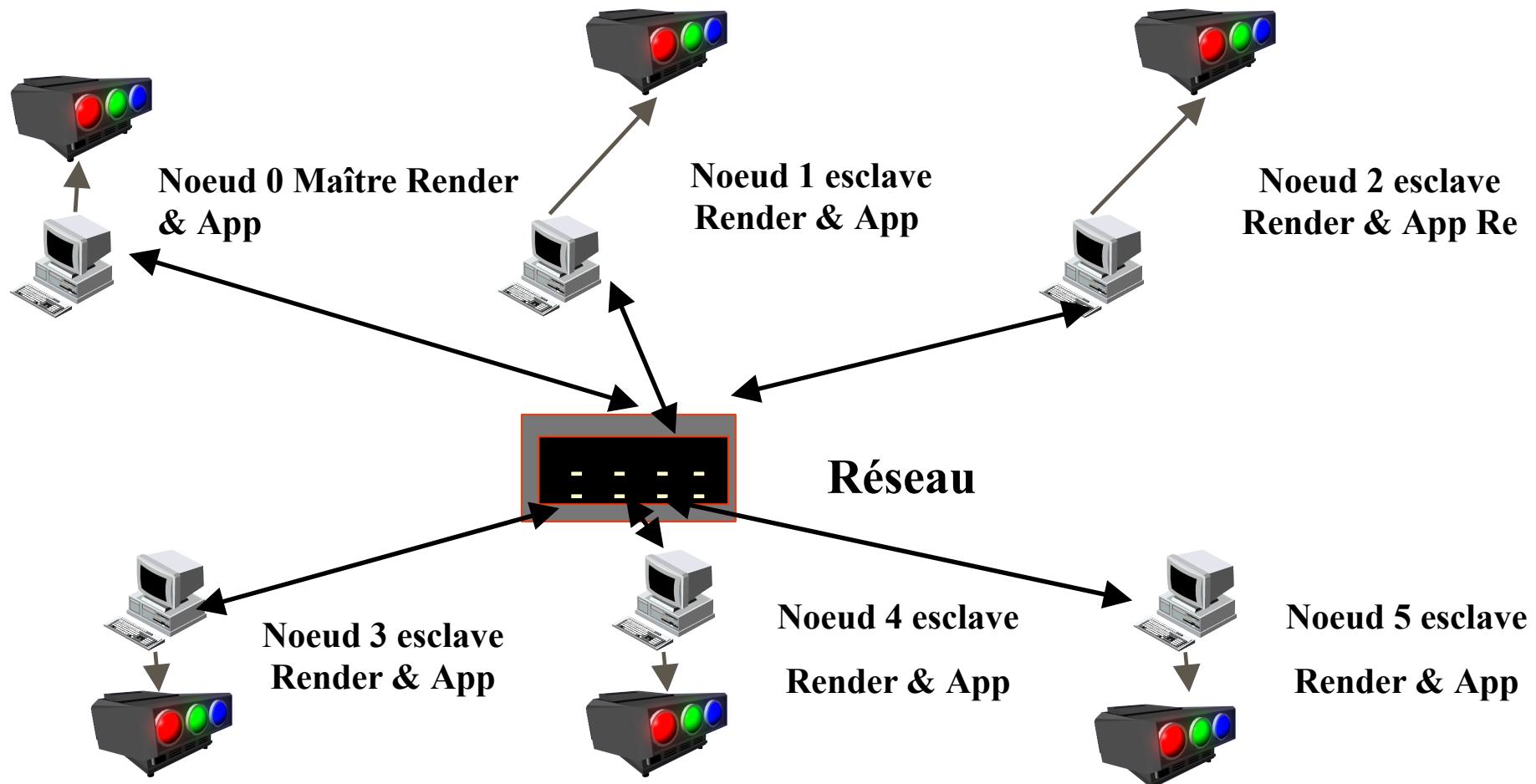
Data-Lock

- Chaque nœud dessine ses frames en utilisant de l'information stockée localement
- Pour que les vues soient cohérentes, les informations sur les nœuds doivent être consistantes
- Solution logicielle seulement.

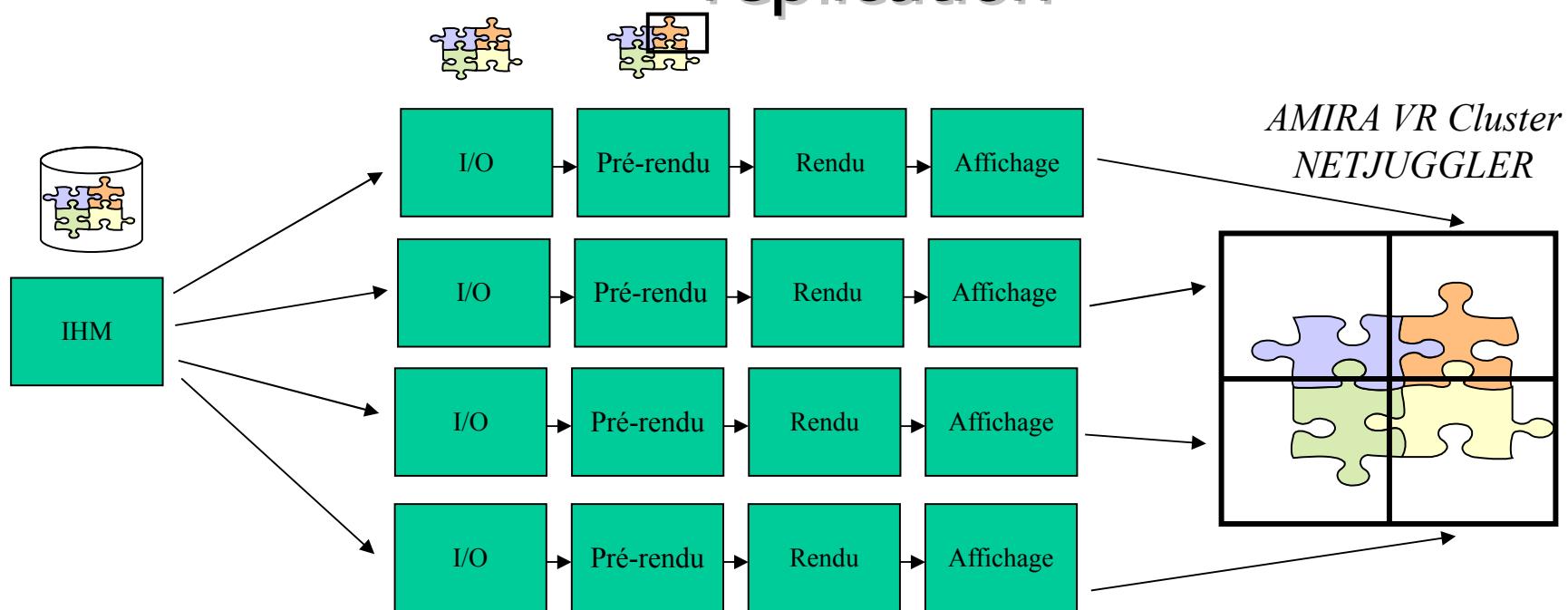
Architecture : Maître/esclave

- Une copie identique de l'application s'exécute sur chaque nœud du cluster
- Une faible quantité de données circule entre les nœuds (données des périphériques d'entrée, temps)
- L'un des nœuds est maître(calcule les changements d'état de l'application et les diffuse vers les autres) et les autres des esclaves
- Peut utiliser un protocole utilisateur

Architecture : Maître/esclave



Distribution des événements de contrôle / réPLICATION

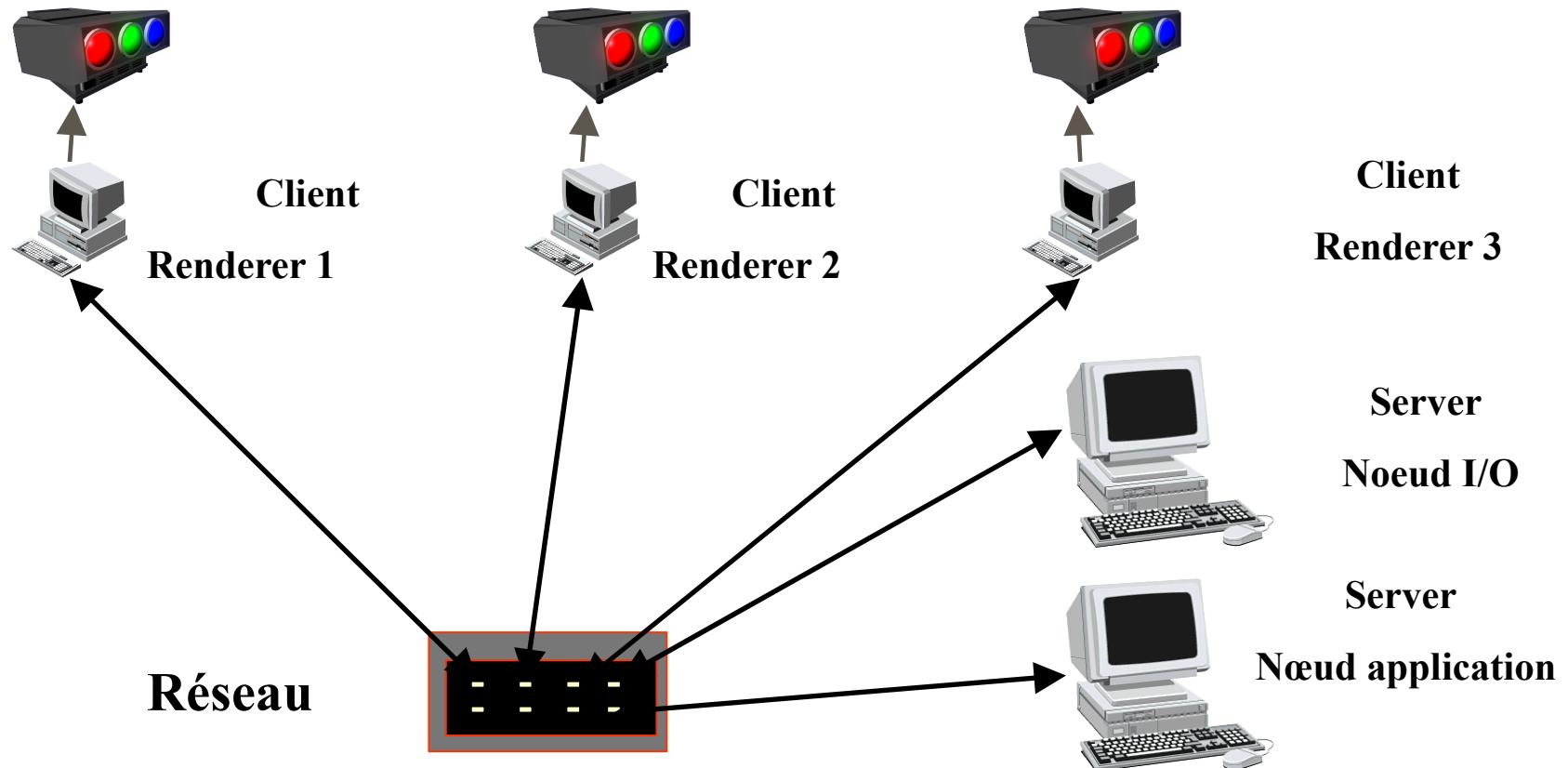


- RéPLICATION de la partie « serveur » ou « esclave »
- Plutôt pour de l'affichage multi-écran
 - $N * \text{nb pixels}$ pour N CPU/GPU dans le même temps
- Pas besoin de réseau haut-débit
- RéPLICATION des données (ou accès commun)
 - Plutôt adapté aux problèmes de taille moyenne et indépendants du temps

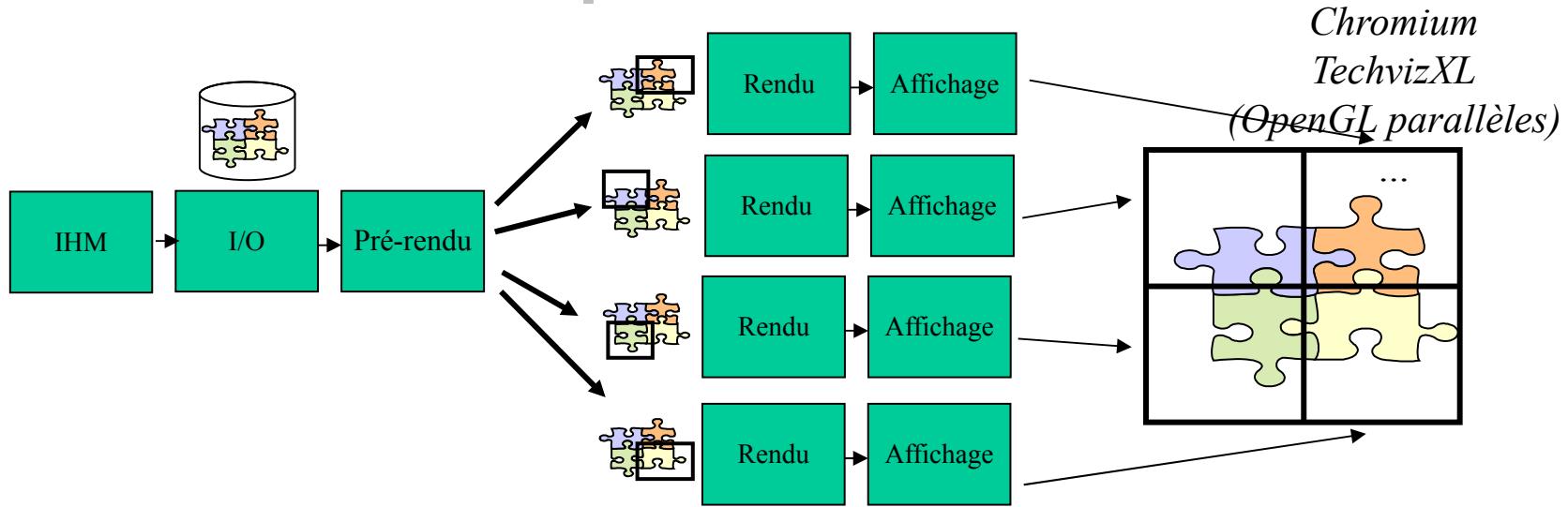
Architecture : Client/Serveur

- Un des noeuds(serveur) alimente les autres (clients) en données.
- Seul le serveur exécute l'application, les clients se bornant à afficher
- Utilise un protocole général.

Architecture : Client/Serveur

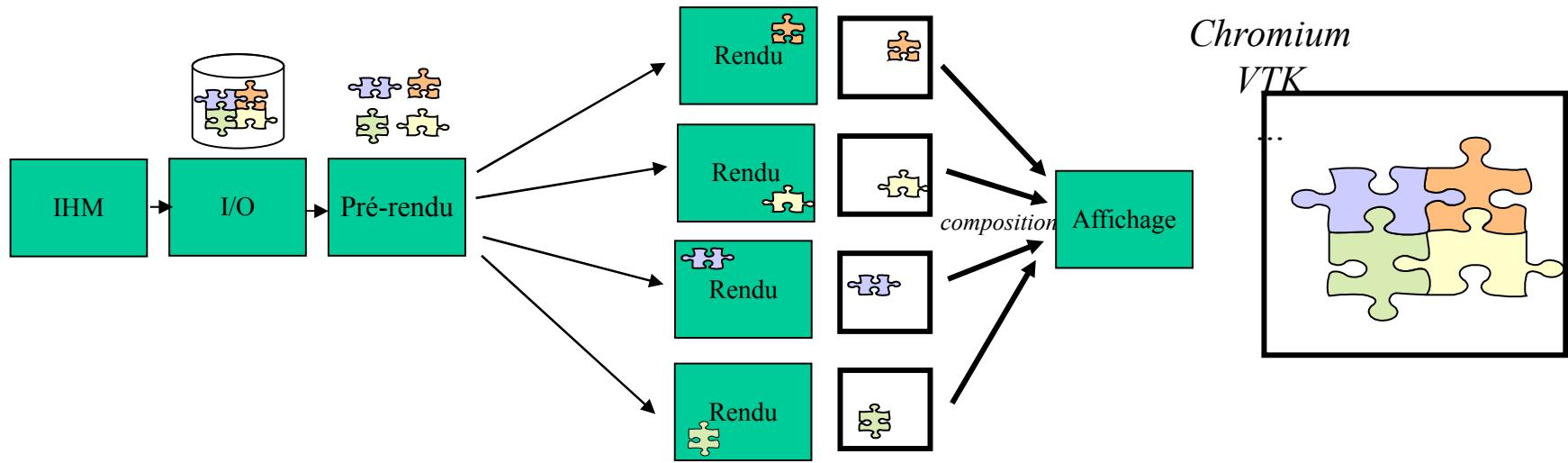


Distribution des primitives exemple en sort first



- Découpage spatial a priori / écran
- 1 GPU = 1 écran ou sous-écran, portion de scène
- Communications plus lourdes
- NB : le schéma précédent est en gros un cas particulier de celui-ci (réPLICATION sans filtreAGE en espace objet)

Composition des pixels (sort last)



- 1 GPU = 1 scène partielle
- Composition différée juste avant affichage : pixel+Z = fragment
- Communications importantes mais bornées à l'avance
 - pas de réplication ni transfert des données => supporte mieux leur croissance en taille
 - taille des images partielles = taille de l'image totale
- Permet de mieux « saturer » les GPU en rendu, voire de leur faire aussi faire readback, composition, compression éventuelle

Architecture : Client/Serveur

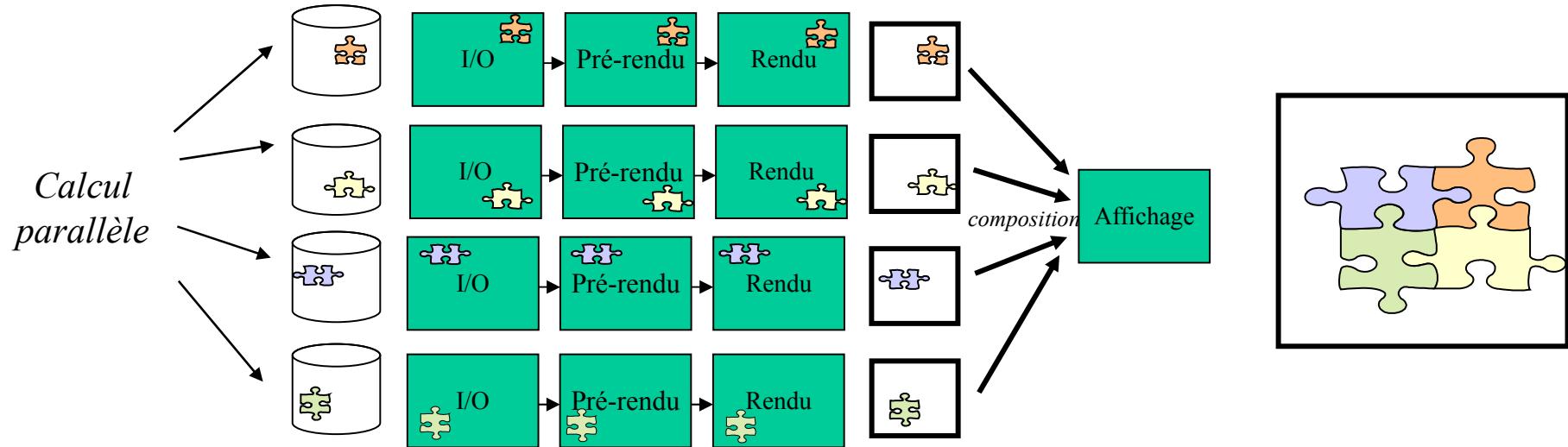
- Peut être plus flexible que l'architecture maître/esclave.
- Mais nécessite souvent une large bande passante

Types d' architectures Client/Serveur

- Quel type de données est utilisé par le protocole?
 - Pixels
 - Primitives graphiques
 - Information de haut niveau telle que données de graphe de scène.
- Le type de données affecte la demande en bande passante

Sort last + pré-rendu parallèle

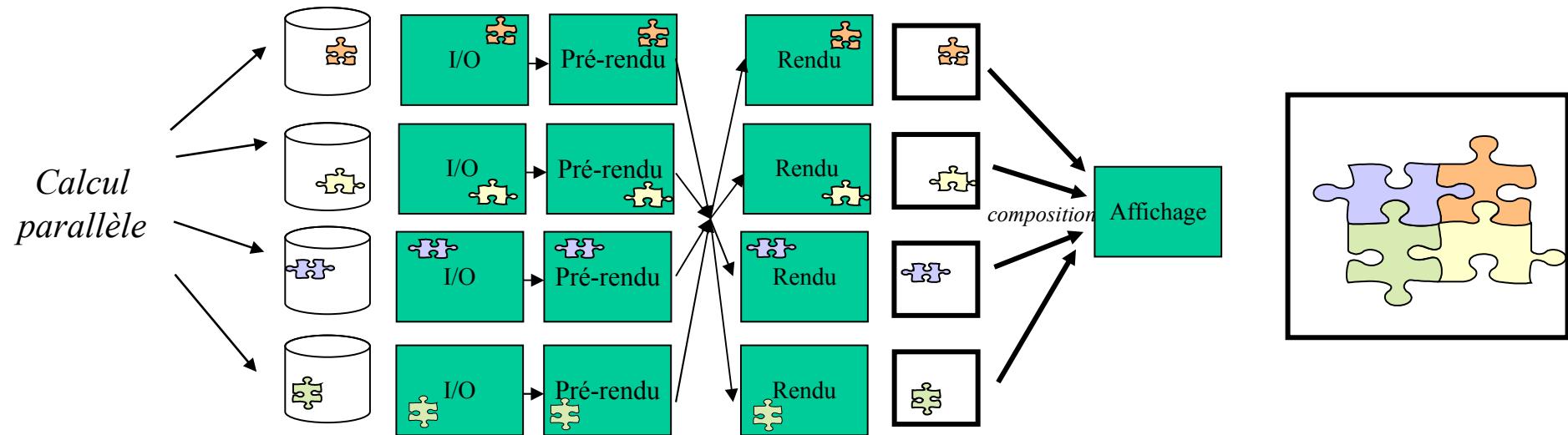
LOVE



- Maintien du parallélisme tout au long du pipeline
- Pas de réorganisation des données
- Il faut quand même traiter les « conditions aux limites des domaines », les problèmes de synchronisation et de cohérence des données affichées.

Variante...

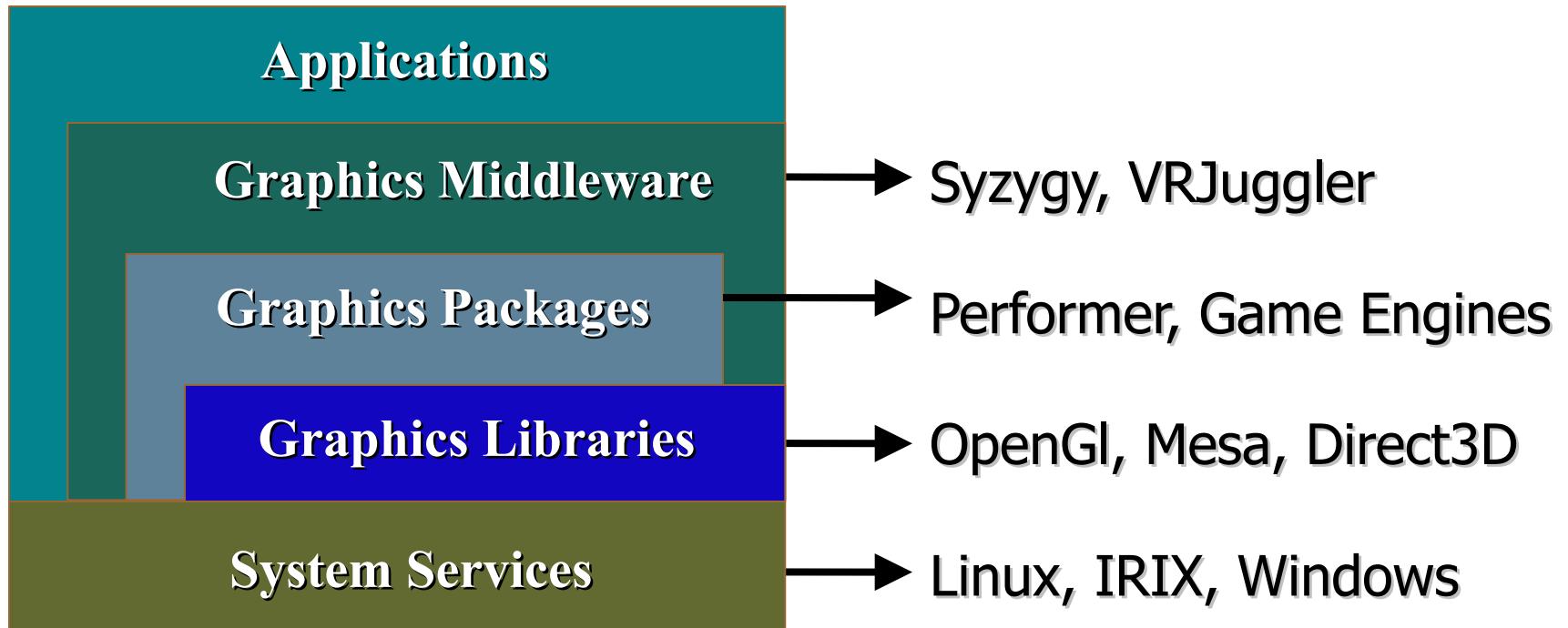
EnSight DR
Paraview



- Ré-équilibrage de charge entre pré-rendu et rendu



Programmation du VR cluster



Programmation du VR cluster

- Portage et développement sur VR Cluster
 - Choisir la bonne stratégie en se basant sur les ressources disponibles (coût/performance)
 - Tel paquetage répond-t-il à mes besoins en RV
 - Est-il facilement modifiable pour marcher sur mon cluster

Programmation du VR cluster

- Considérer l'héritage logiciel
 - 10 ans de développement sur les machines haut de gamme
- Besoin de considérer la convergence

High End:

OpenGL

Applications

High-end App



VR Cluster

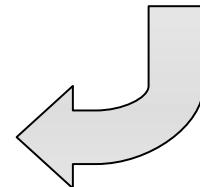
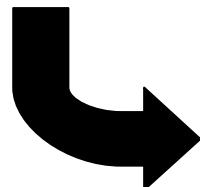


PC:

Direct3D

Games

Low-end App



Programmation du VR cluster

librairies graphiques

- OpenGL, Direct3D

Paquetages graphiques

- Performer, Game Engines, VTK

plateformes VR

- VR Juggler, CaveLib, Diverse, MultiGen-Paradigm

Programmation du VR cluster

Protocoles réseau

- ♦ TCP/IP, VIA, GM

Paquetages cluster

- ♦ MPI, OpenPBS, Clic

Middleware de communication

- ♦ CORBA, VNC

Programmation du VR cluster

librairies VR Cluster

- ♦ WireGL, SoftGenLock, DICELib

Paquetages VR Cluster

- ♦ OpenPBS

Middleware VR Cluster

- ♦ Syzygy, NetJuggler



Exemple de déploiement sur un Cluster de RV

Souley Madougou (BRGM/LIFO - Orléans University)

Jacques Vairon (BRGM - French Geological Survey)



Laboratoire d'Informatique
Fondamentale d'Orléans



Hardware

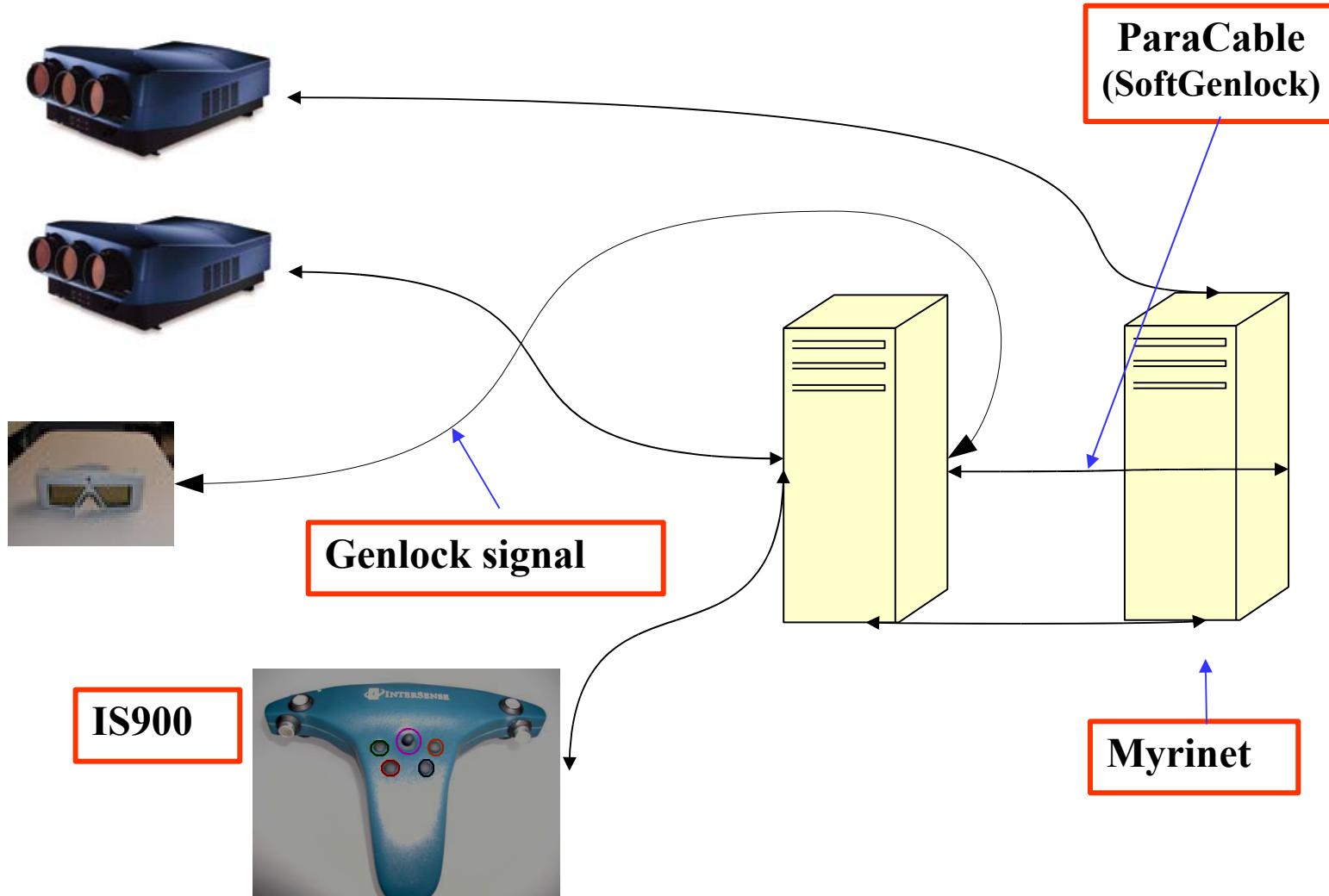


Cluster de 6 nœuds

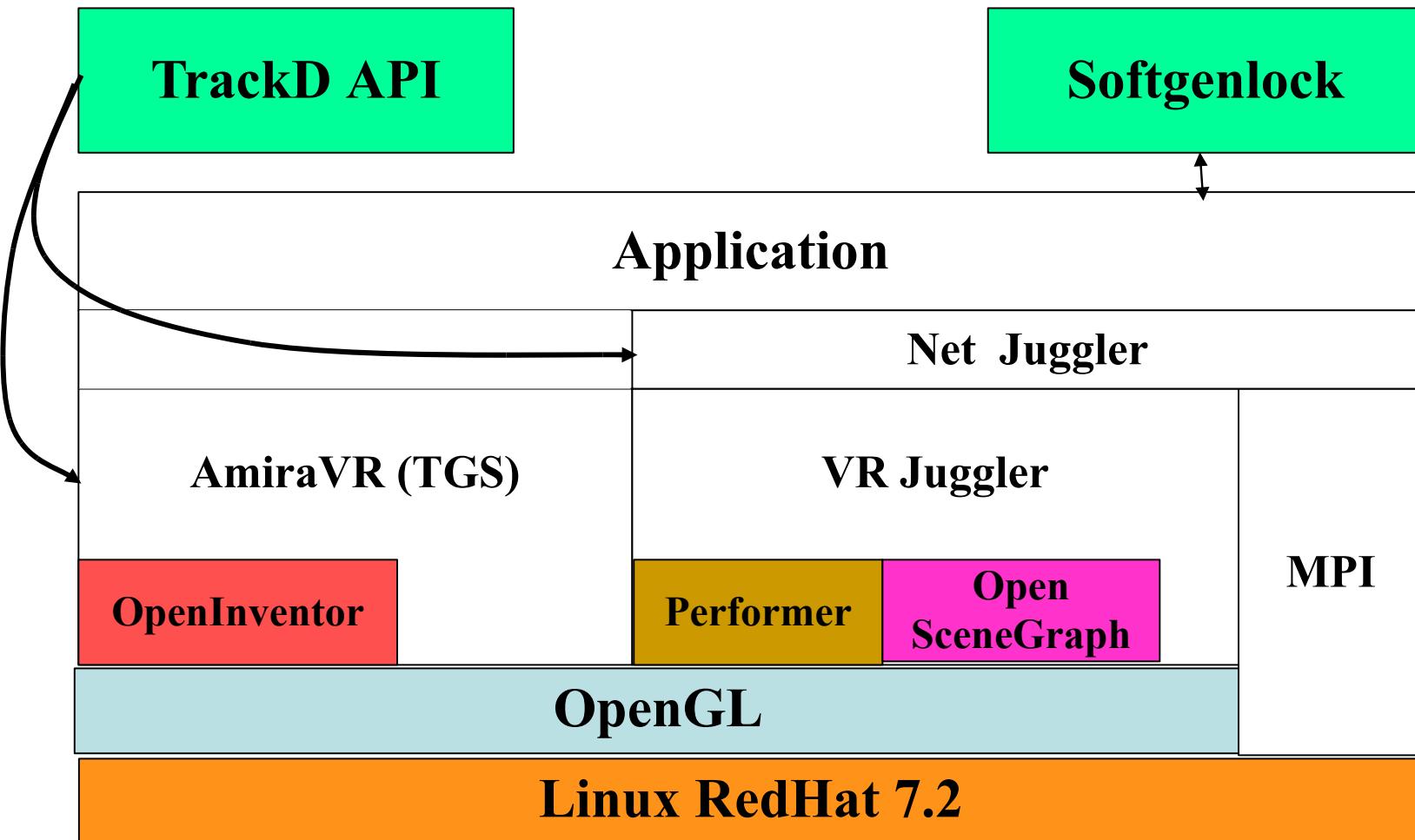


Barco Consul workbench

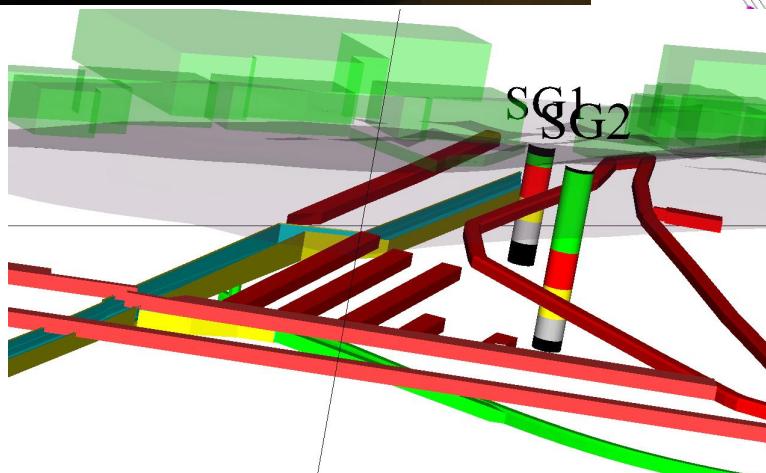
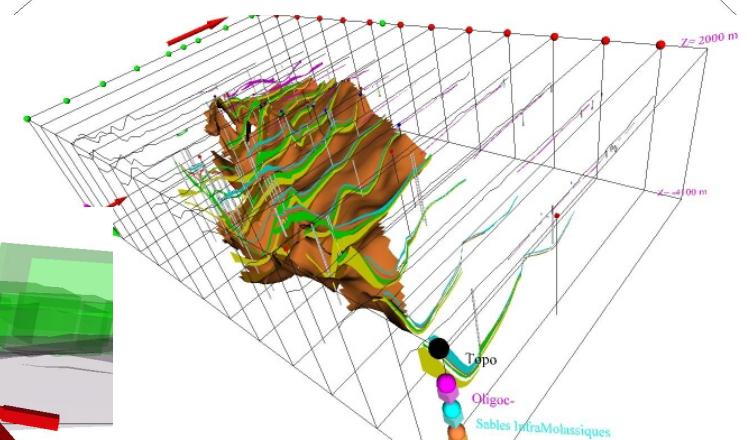
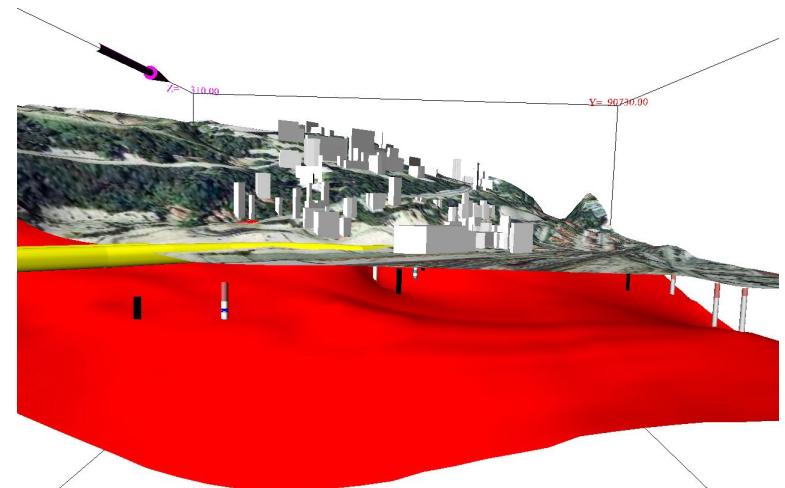
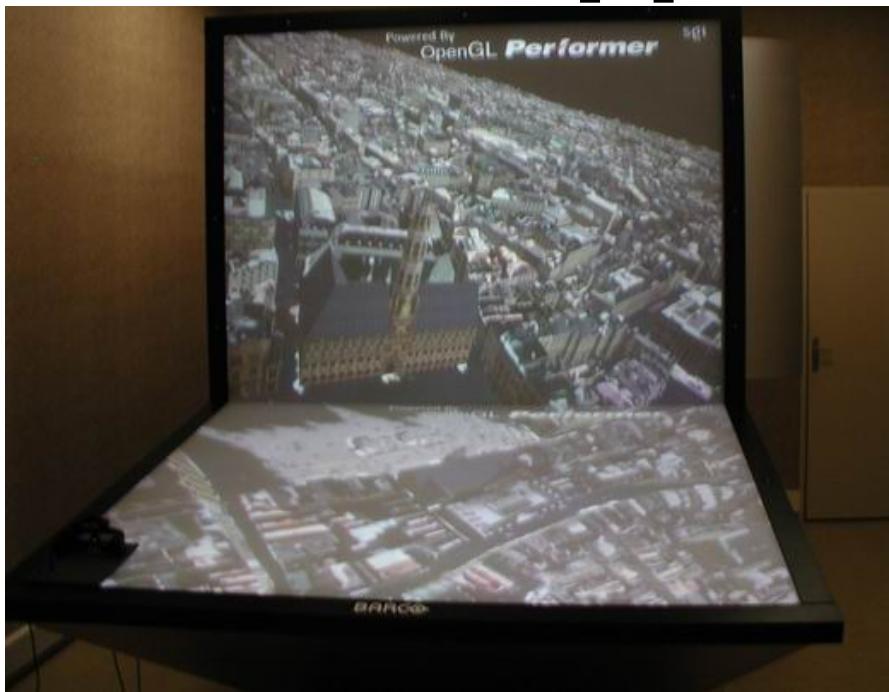
Hardware setup



Software



Applications



Net Juggler

- **Net Juggler duplique l'application:**

Peu de données de communication mais duplication des données et calculs

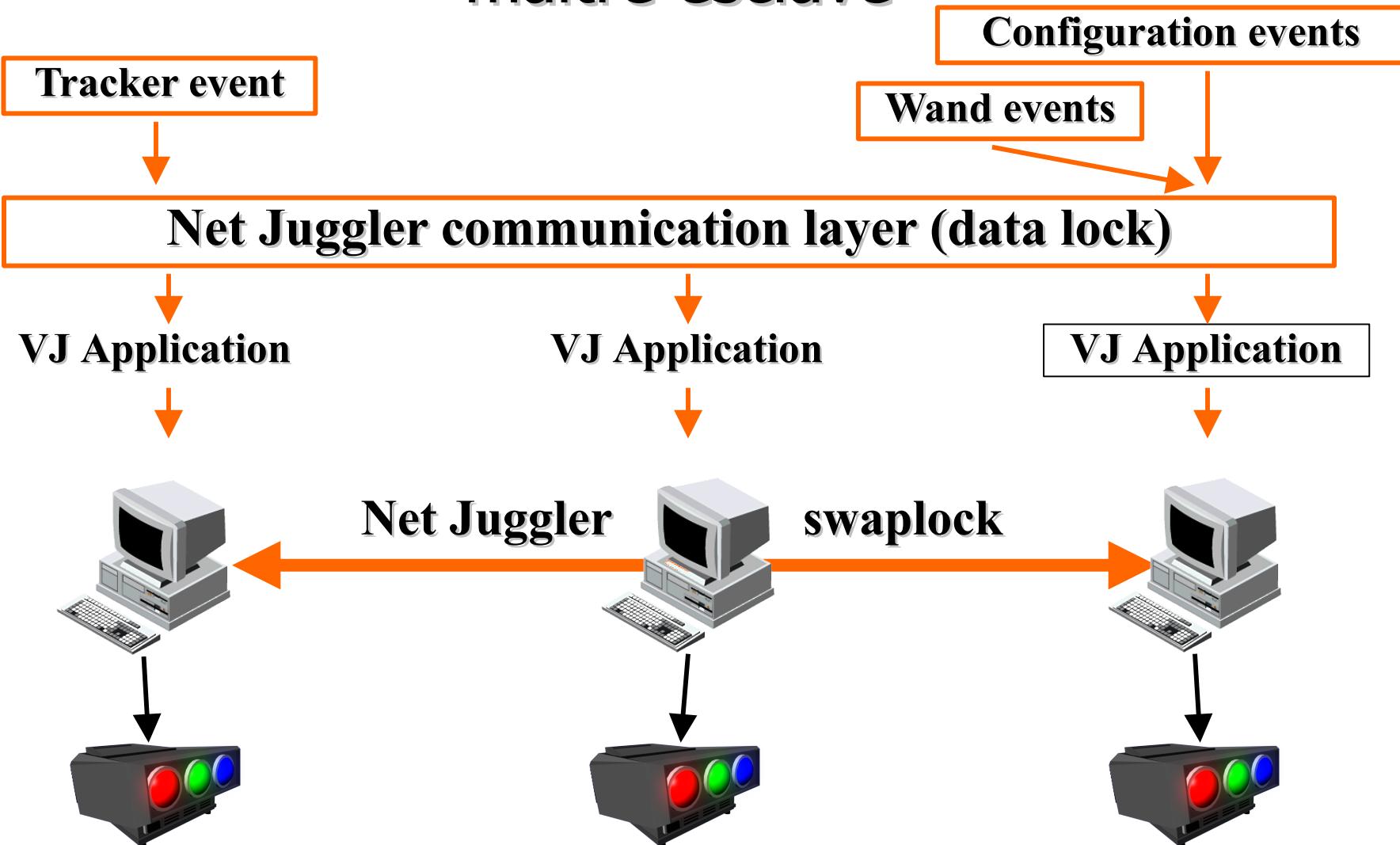
Ok pour 80% des applications

fps sur un PC = fps sur 8 PC du cluster

- **Dans certains cas, la performance peut être améliorée en distribuant les données et les calculs**

Surveiller le coût des communications supplémentaires

Net Juggler: parallélisation de type maître esclave



Net Juggler



Lancement d'applications

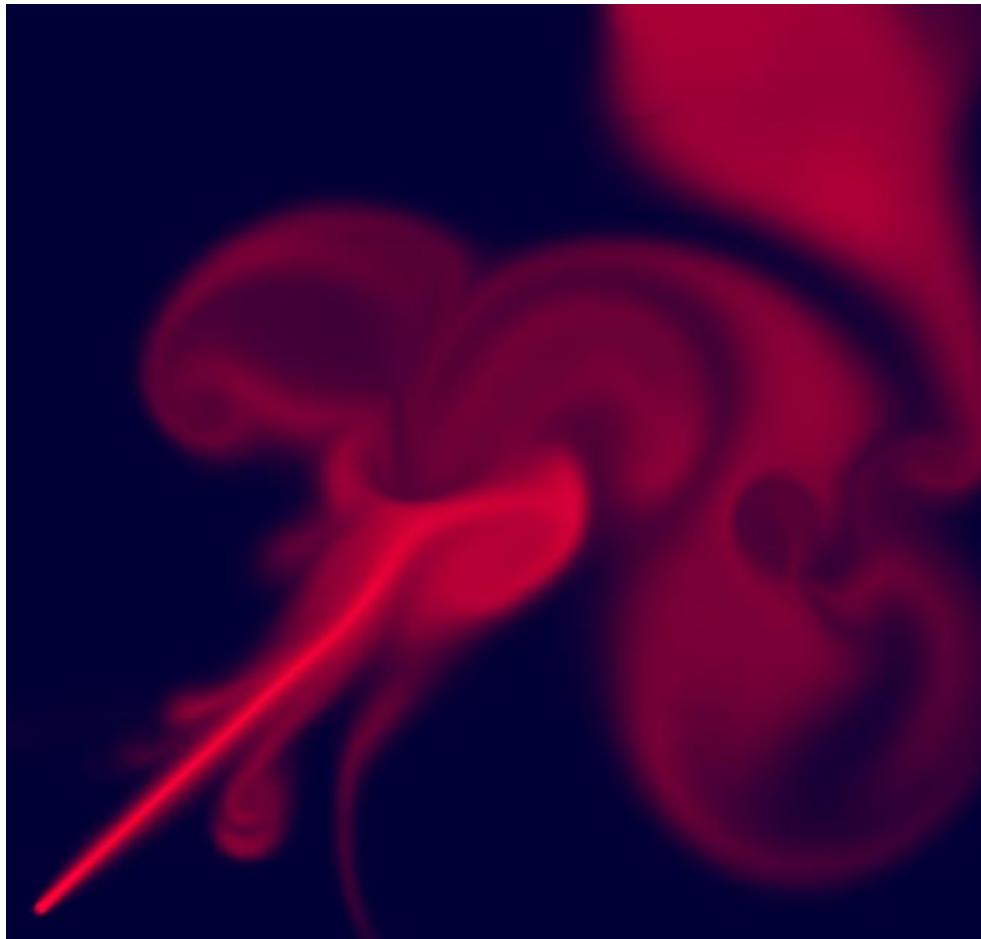


Gestion des gros volumes

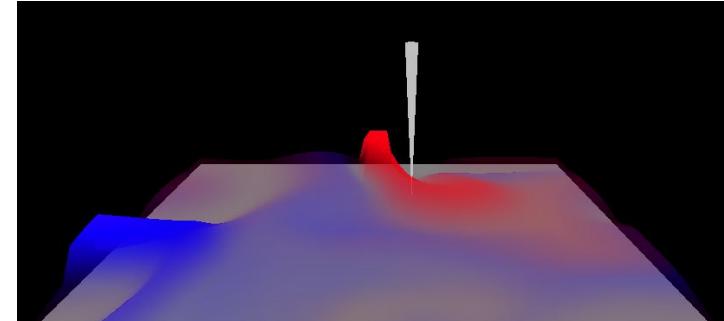
- Cache des données
- Niveaux de détails



L'application Fluid



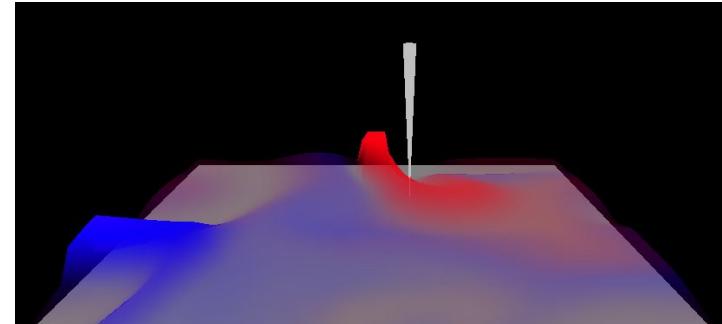
L'application Fluid



- **Simulation de fluides: solveur des équations de Stam Navier-Stokes**
- **Grille 2D distribuée sur les noeuds + solveur parallèle des équations de Navier-Stokes**
- **Implémentation utilisant la librairie mathématique PETSc (basée sur MPI) ou solveur « multigrid »**
- **Net Juggler: plateforme pour rendu et pré-rendu distribués**

Résultats

- **6 dual PIII-800 MHz 2D**
- **GeForce 2 GTS 64 MB**
- **100 Mbits/s Fast Ethernet Network**
- **128x128 grid**



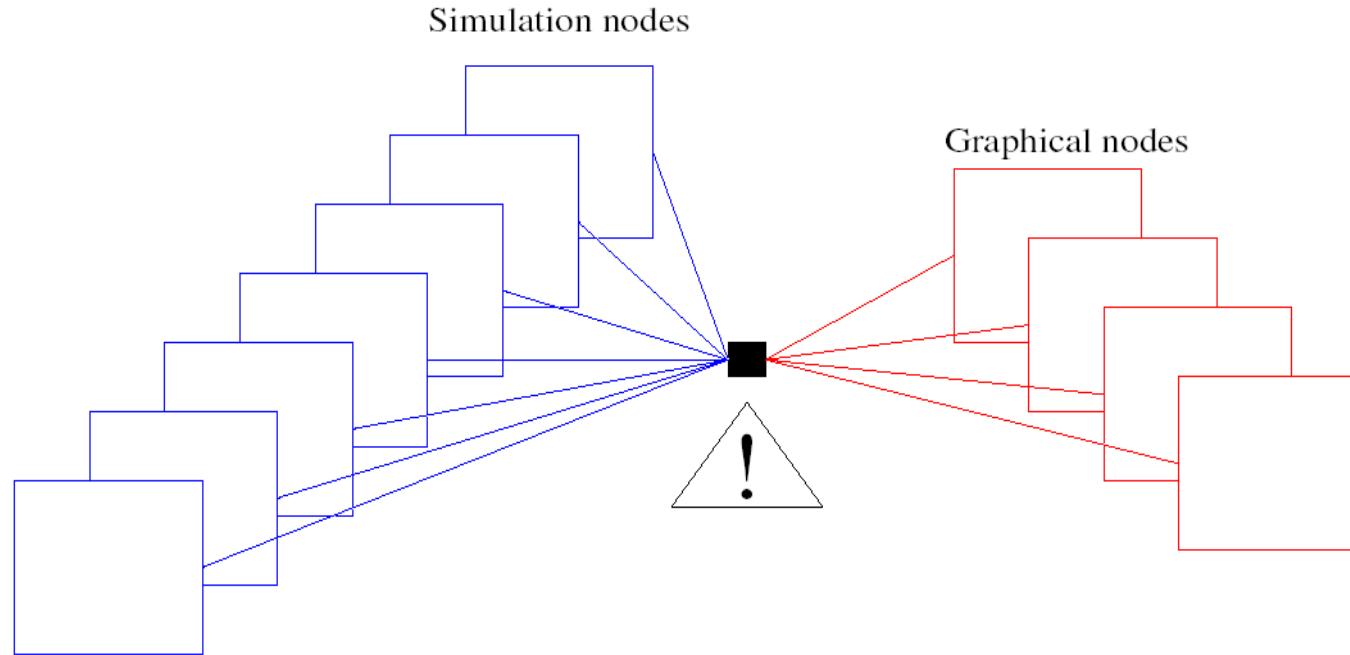
	1 écrans	4 écrans
Solveur dupliqué	8 fps	8fps
Solveur distribué sur 4 noeuds	21 fps	20 fps
Solveur distribué sur 6 noeuds	24 fps	22 fps

Architecture de l'application



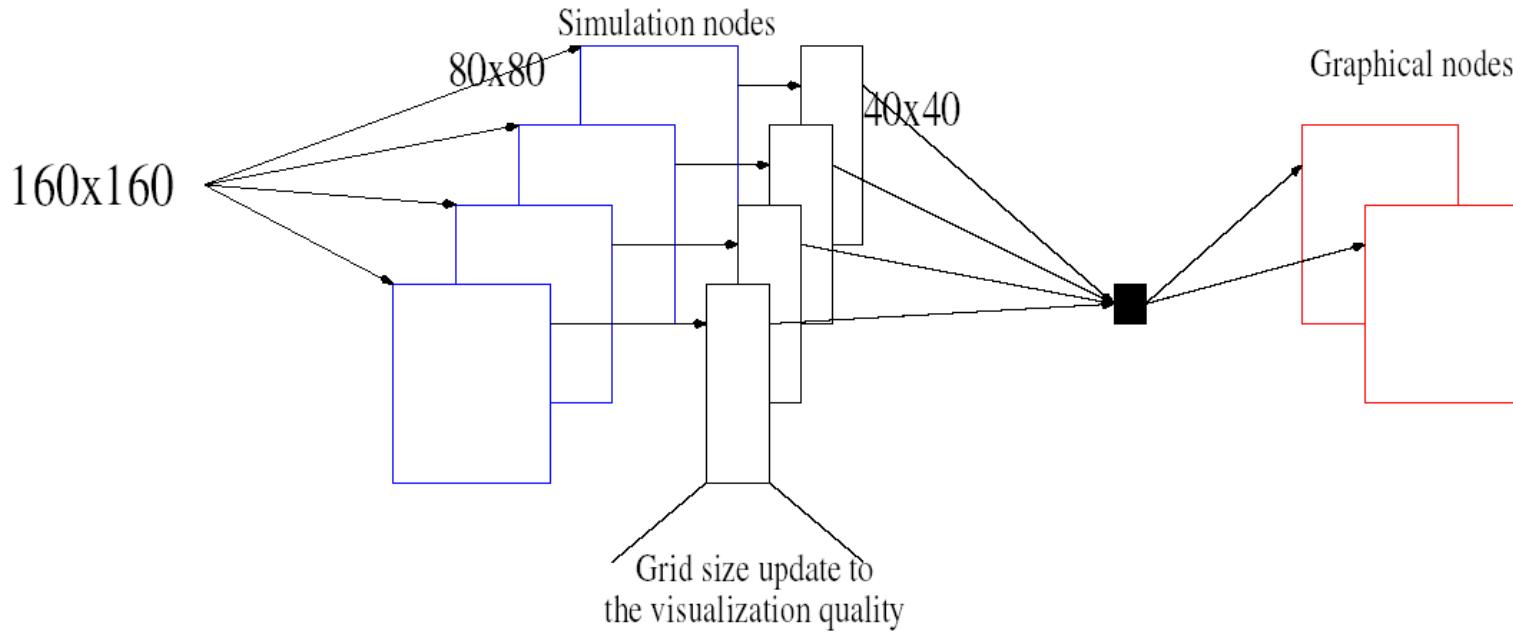
- Spécialisation des noeuds
- Utilisation du swaplock comme barrière de synchronisation

Couplage de la simulation et de la visualisation



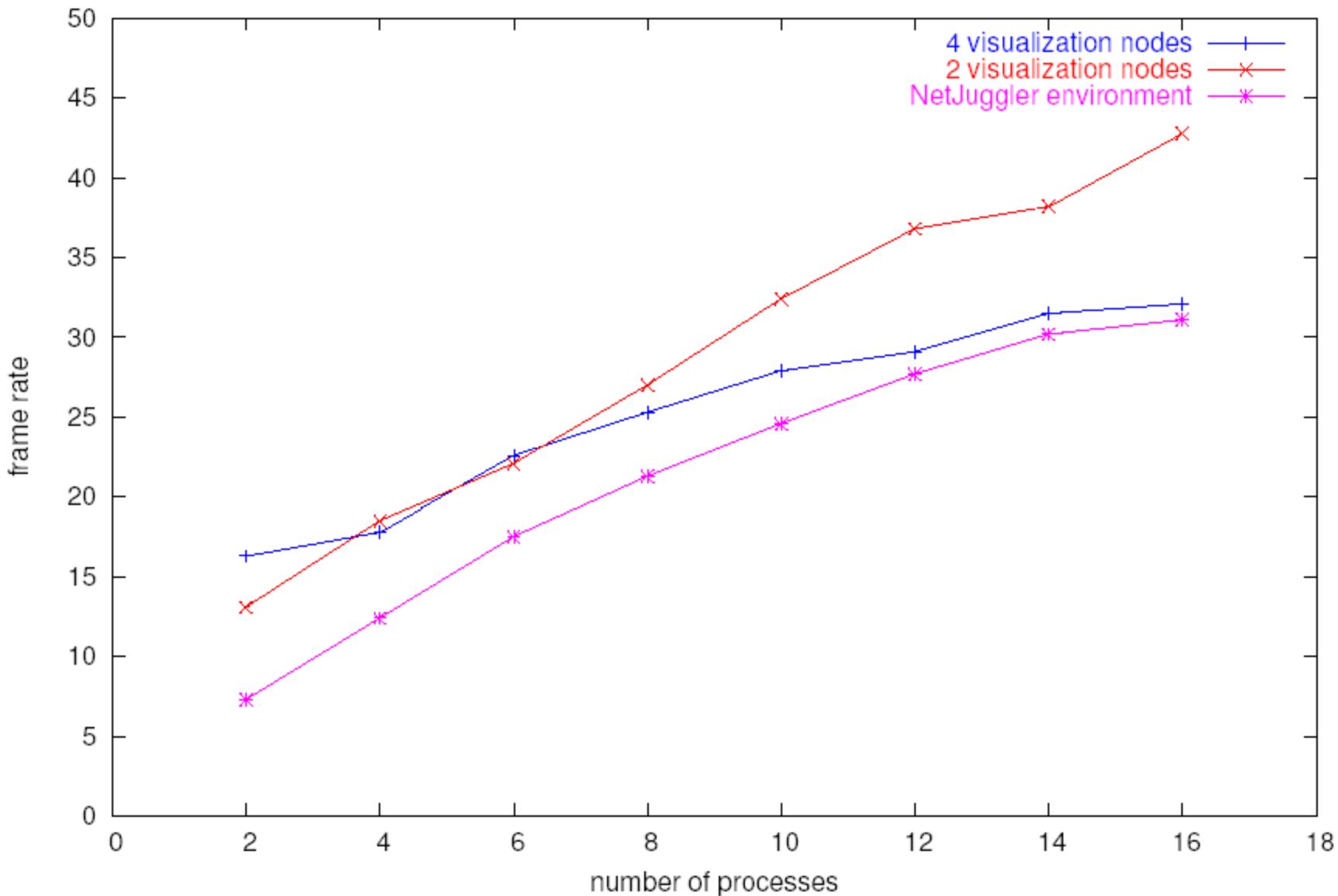
- Peut rapidement devenir un goulot d'étranglement
- Peut être source d'optimisations si on distingue les contraintes de la simulation et de la visualisation

Couplage de la simulation et de la visualisation



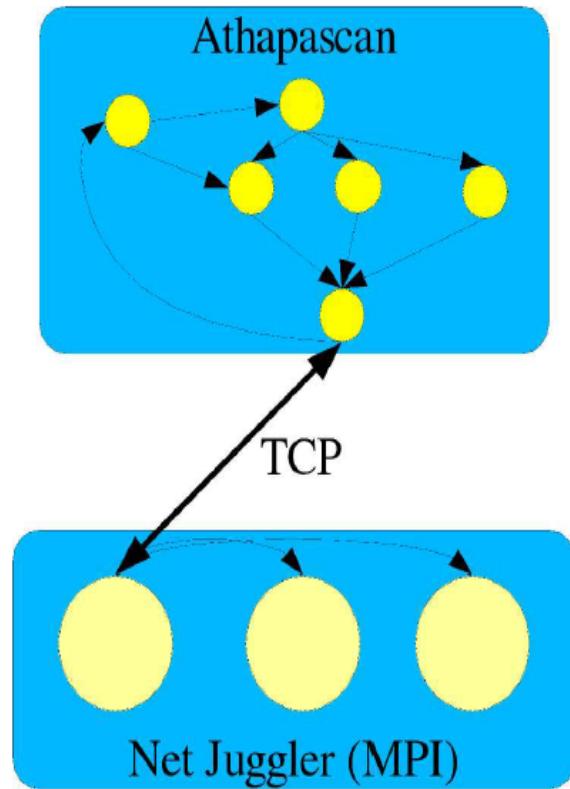
- NetJuggler est l'environnement de développement
- Un programme MPI contrôle tout le flot de données
- Des processus additionnels entre le rendu et le pré-rendu

Frame rate from number of nodes



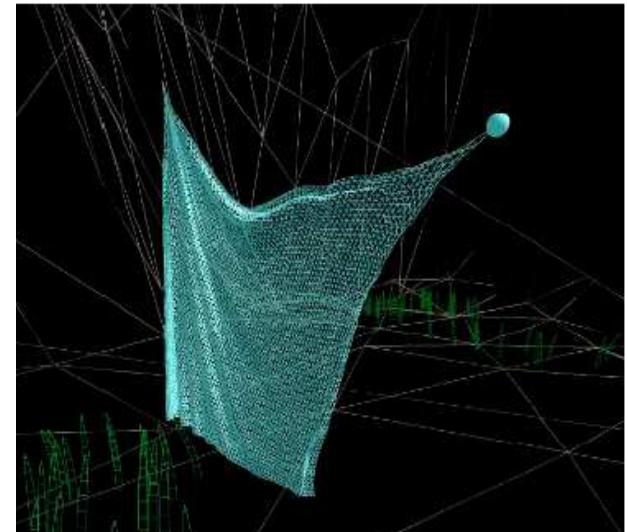
Une autre expérience de couplage

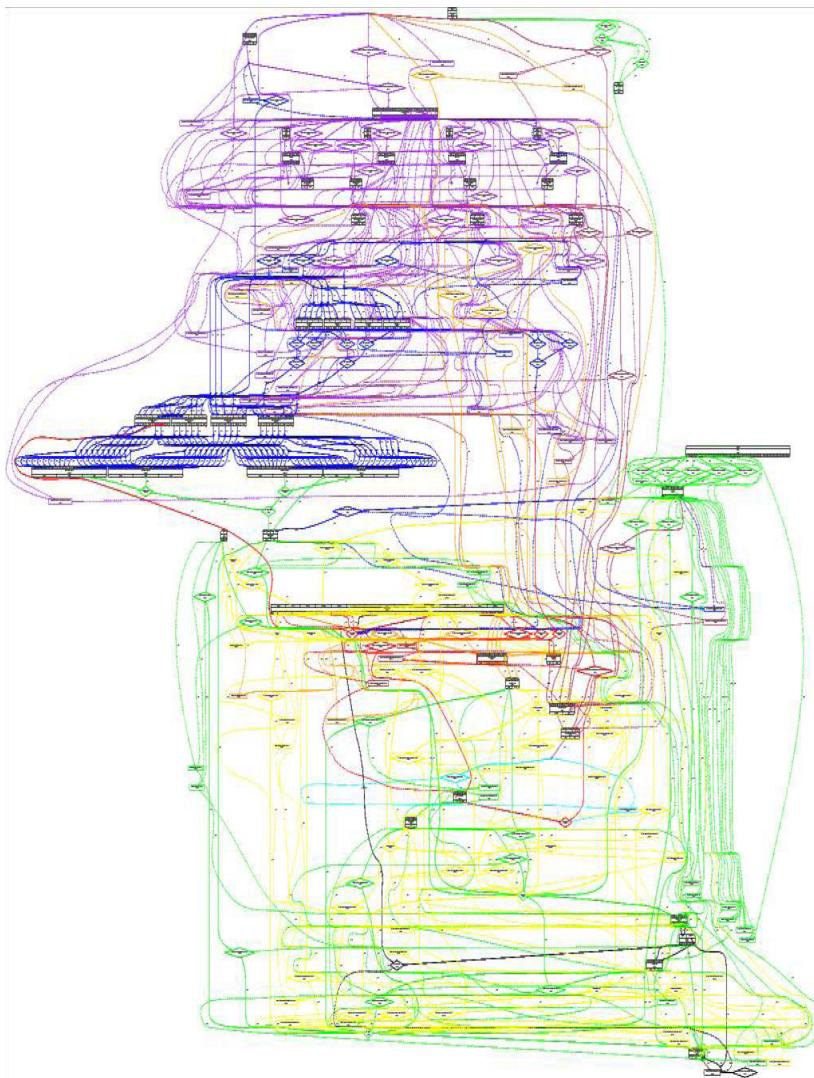
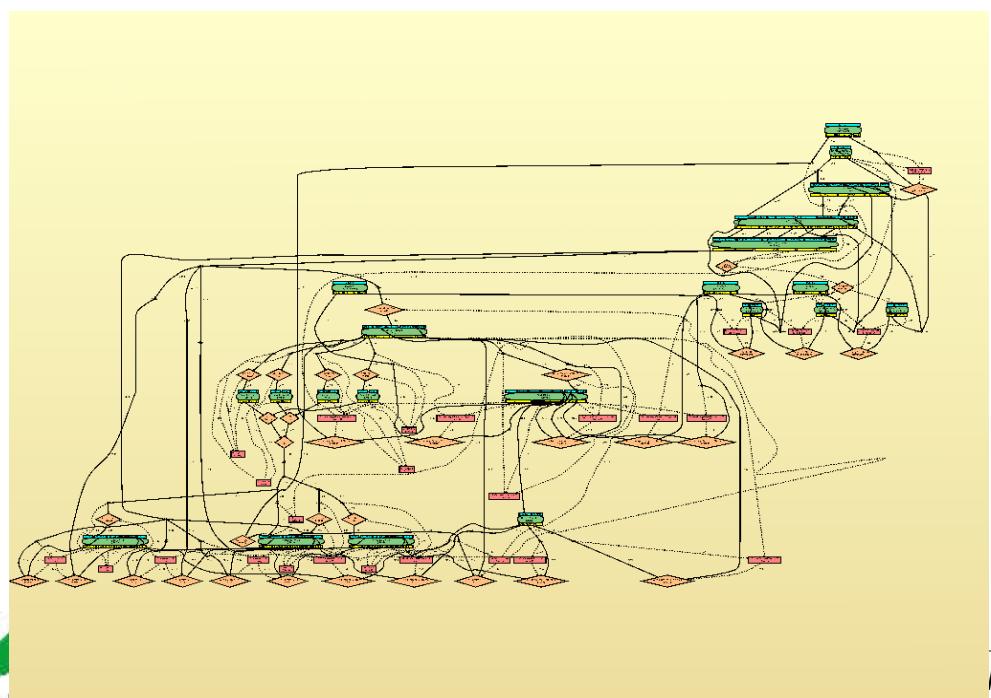
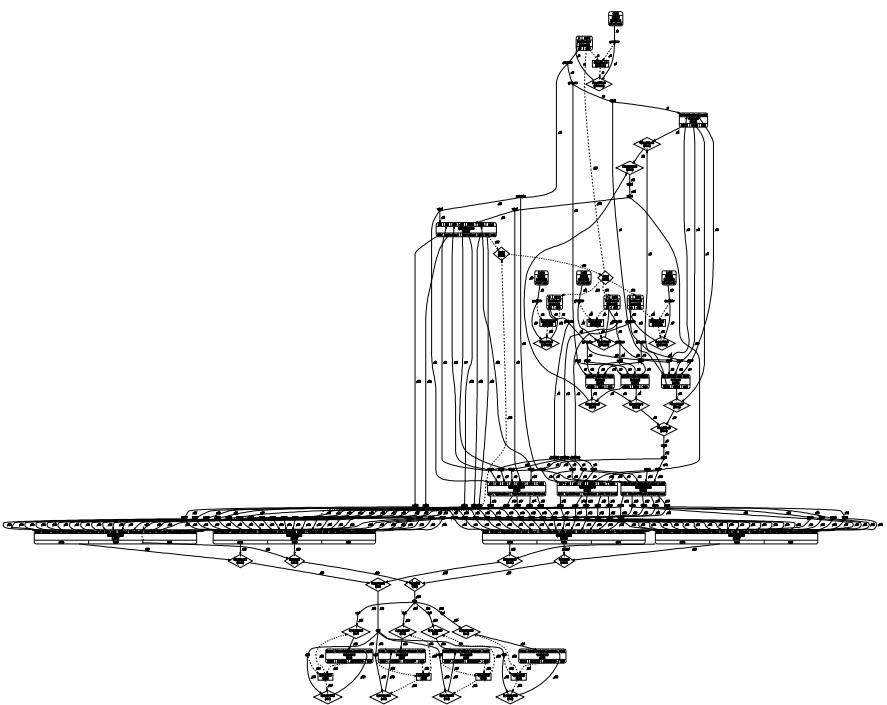
- J. Allard, B. Raffin et F. Zara, Euro-Par 2003
- Simulation de tissus avec Athapascan
- Communication des données via TCP
 - Centralisation
- Asynchronisme possible
 - La visualisation lit les derniers résultats disponibles



Bilan

- Couplage d'une visualisation multi-écrans avec plusieurs simulations distribuées
- Performances suffisantes pour l'interaction
- Code spécifique pour cette application
- Problème de passage à l'échelle (centralisation)
- Pas de contrôle de cohérence
- Pas de support pour la mise au point (niveau de détail, volume de données, latence, synchronisations complexes etc...)
- Pas de support pour le couplage de codes hétérogènes





4 cameras,
50 processeurs, 16
projecteurs.