

Amélioration de la réactivité des réseaux pair à pair pour les MMOGs

Xavier Joudiou,
Encadré par: S.Legtchenko & S.Monnet

Université Paris VI, Master SAR

8 Septembre 2010



1 Introduction

2 État de l'art

- Solipsis
- Les traces
- BlueBanana

3 Les améliorations

4 Le cache pour les zones denses

- Explications du cache pour les zones denses
- Résultats pour le cache
- Conclusion sur le cache des zones denses

5 Le préchargement amélioré des données

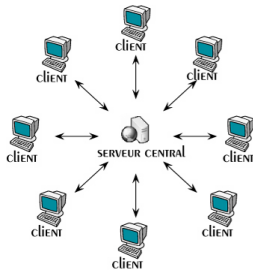
- Explications du préchargement amélioré
- Résultats pour l'amélioration du préchargement des données
- Conclusion sur l'amélioration du préchargement des données

6 Conclusion

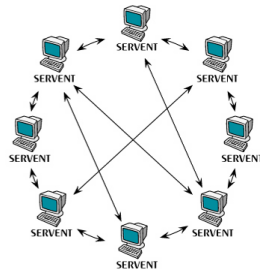
Présentation des points importants à la compréhension du sujet :

- Architecture pair à pair Vs Client-Serveur
- Définition de l'overlay

Architecture pair à pair Vs Client/Serveur



ARCHITECTURE CLIENT-SERVEUR

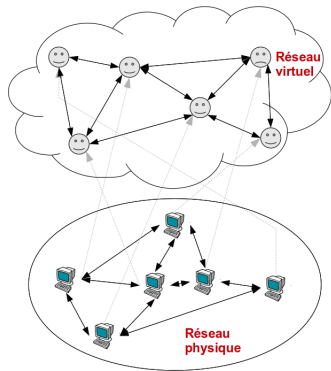


ARCHITECTURE PAIR-À-PAIR

- Problème du passage à l'échelle de l'architecture Client/Serveur.
- Solutions p2p existantes pas assez réactive pour assurer une latence suffisante.

Définition de l'overlay

- Un overlay est un réseau informatique formant un graphe où les liens sont déterminés avec un critère logique.
- Réseau physique \neq Réseau virtuel



Présentation des mécanismes importants à la compréhension des solutions introduites :

- Solipsis
- Étude des traces des joueurs de MMOG
- Blue Banana



Solipsis :

- propose un monde virtuel entièrement décentralisé et scalable.
- met en place un overlay avec une forte malléabilité applicative.
- chaque entité collecte des informations pour reconstituer son environnement local.

Solipsis introduit plusieurs propriétés :

- *Local Awareness* :

Une entité doit être connectée avec tous ses plus proches voisins, elle peut connaître des entités en dehors de son environnement virtuel local. Toute entité située à l'intérieur de l'environnement d'une entité doit faire parti des voisins de cette entité.

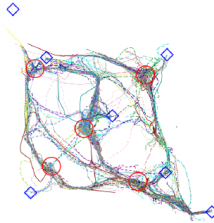
- *Global Connectivity* :

Toute entité virtuelle doit se trouver à l'intérieur de l'enveloppe convexe contenant l'ensemble de ses voisins logiques.



Des études des traces des joueurs de MMOG, ont permis de détecter différents habitudes des joueurs :

- Présence de zones denses
- Mouvements erratiques dans les zones denses
- Mouvements rectilignes et rapides entre les zones denses



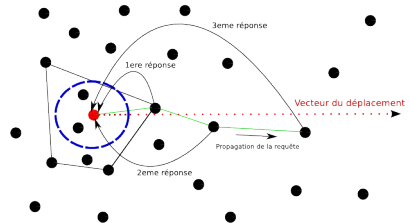


Blue Banana introduit trois états, pour un avatar :

- **H**(alted) : l'avatar est immobile.
- **T**(ravelling) : l'avatar se déplace rapidement sur la carte et il a une trajectoire droite.
- **E**(xploring) : l'avatar est en train d'explorer une zone, sa trajectoire est confuse et sa vitesse est lente.

Mise en place d'un mécanisme d'anticipation des mouvements des avatars.

- Si état **T**, il cherche des nœuds sur sa trajectoire.
- Evaluation du nœud, propagation de la requête.
- Réponse au nœud qui a demandé le préchargement.



Durant le stage, plusieurs solutions ont été implémentées :

Durant le stage, plusieurs solutions ont été implémentées :

- Le cache pour les zones denses

Durant le stage, plusieurs solutions ont été implémentées :

- Le cache pour les zones denses
- Le préchargement amélioré des données

Durant le stage, plusieurs solutions ont été implémentées :

- Le cache pour les zones denses
- Le préchargement amélioré des données

D'autres solutions ont été étudiées, mais sans être implémentées.

- Mouvements de groupe
- Connaissance des routes entre les zones denses

Différentes métriques utilisées pour analyser les résultats :

- Nombre de messages à un instant
- Cohérence de la topologie
Nombre de nœuds dans la zone de connaissance mais pas dans l'ensemble des voisins
- Cache Hit et Miss (pour le cache)

En fonction du degré de mobilité.

Le cache des zones denses

- Explications sur le cache des zones denses
- Les résultats
- Conclusion sur le cache

Comment fonctionne le cache ?

- Chaque nœud de l'environnement a un cache.
- Il est utilisé seulement par les nœuds en état **E**xploring).
- Deux types de cache mis en place (retour simple et retour multiple).

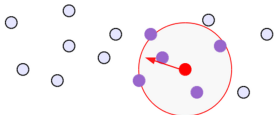
Trois types de recherche dans le cache :

N	Critère de sélection	Avantages	Inconvénients
1	Comparaison distances	Simplicité	Distance \neq utile, aide pas enveloppe
2	Aide enveloppe	+ Enveloppe OK	- bon règles Solipsis
3	Zone de connaissance	Simplicité	aide pas enveloppe

- La version 3 a été conservé pour les tests finaux.



Explications du cache pour les zones denses



1^{re} phase : le nœud rouge a ses voisins (en violet) et se prépare à se déplacer



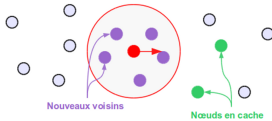
Liste des voisins



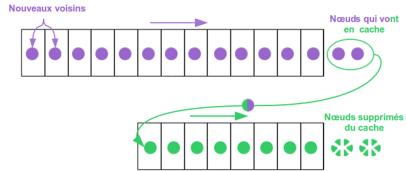
Cache



Explications du cache pour les zones denses

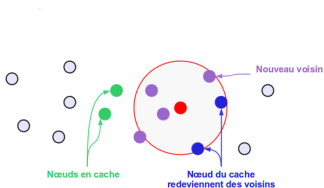


2^e phase : le nœud rouge a des nouveaux voisins, des nœuds en cache (en vert), et se prépare à se re-déplacer

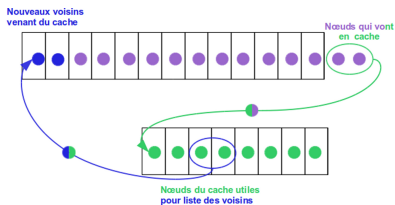




Explications du cache pour les zones denses



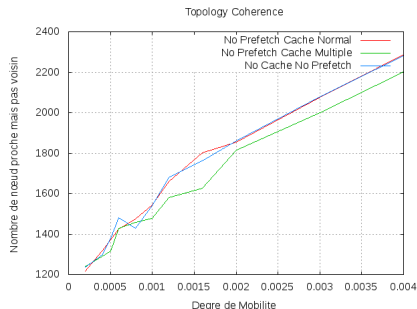
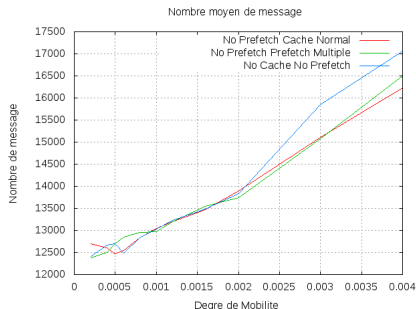
3^e phase : le nœud rouge a des voisins, des nœuds en cache et d'anciens nœuds du cache redeviennent ses voisins (en bleu)



Différents mécanismes pour le cache :

- Mise à jour des données du cache
- Contact un nœud du cache s'il est là depuis longtemps
- Aide les nœuds voisins lors de recherche de nœud

Nombre de messages et Cohérence de la topologie



Résultats des deux versions du cache en fonction d'une version sans le cache activé :

Solution	Cohérence topologie	Nombre de messages
Cache simple	Équivalente	5% de msg en moins
Cache multiple	3% de gains	5% de msg en moins

- Moins de messages le cache s'utilise immédiatement sans message.
- Gain sur la cohérence de la topologie si retour multiple.

La mise en place du cache permet :

- d'économiser des messages.
- d'améliorer la cohérence de la topologie.

Amélioration possible en testant toutes les combinaisons de paramètres (mise à jour, contact d'un nœud, taille du cache, etc).

Le préchargement amélioré des données

- Explications sur le préchargement amélioré
- Les résultats
- Conclusion sur le préchargement

Situation : Le préchargement de Blue Banana prend tous les nœuds, à bonne distance, dans le cône.

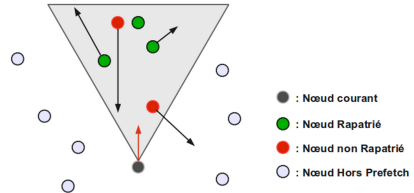
Problème : Des nœuds inutiles sont préchargés.

Solution : Choisir plus finement les nœuds qui vont être sélectionnés.

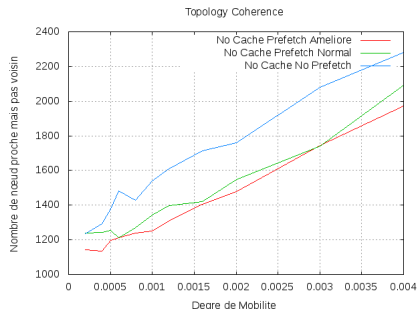
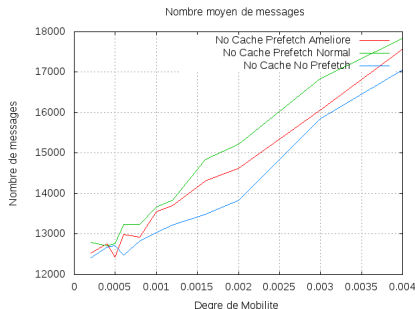
Comment : Regarder la direction des nœuds et leur vitesse.

Préchargement si :

- l'angle du nœud est proche du nœud courant
- *Somme des normes des vecteurs \geq Norme du vecteur de prefetch $\pm \Delta$*
- l'angle du nœud n'est pas proche du nœud courant mais sa norme est inférieure à celle du nœud courant



Nombre de messages et Cohérence de la topologie



Résultats des différents préchargements en comparaison à une version sans le préchargement activé :

Solution	Cohérence topologie	Nombre de messages
Normal	15% de gains	8% de msg en plus
Amélioré	16% de gains	4% de msg en plus

- Gain en terme de messages car préchargement plus efficace, et donc moins de recherche de voisins.
- Léger gain sur la cohérence de la topologie car élimination du préchargement de certains nœuds inutiles.

Notre amélioration du préchargement permet :

- d'économiser des messages par rapport au préchargement normal.
- d'améliorer légèrement la cohérence de la topologie.

Possibles améliorations du préchargement en regardant d'autres paramètres, comme la distance avec les nœuds.

- Les solutions ont permis d'améliorer la réactivité des réseaux pair à pair pour les MMOGs.
- Meilleure cohérence de la topologie et moins de message que dans Blue Banana.
- Autres pistes d'amélioration trouvées :
 - Meilleure utilisation des mécanismes du cache
 - Amélioration du préchargement
 - Mouvements de groupe
 - Route entre les zones denses.

Merci.

Questions ?