

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

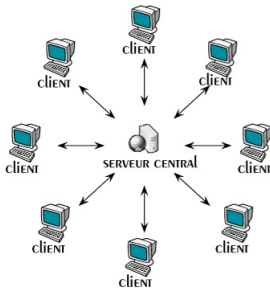
- Explications du préchargement amélioré
- Résultats pour le préchargement amélioré
- Conclusion sur le préchargement amélioré

6 Conclusion

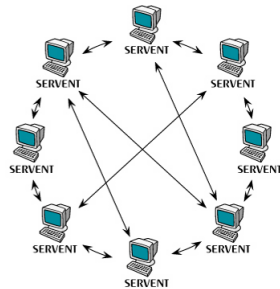
Présentation des points importants à la compréhension des solutions :

- Architecture pair à pair Vs Client-Serveur
- Définition d'un Overlay

Architecture pair à pair Vs Client/Serveur



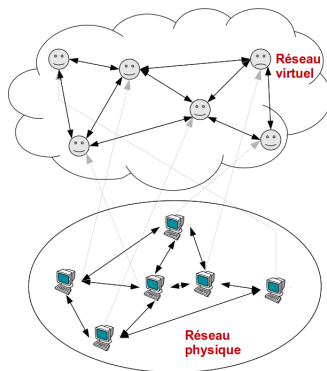
ARCHITECTURE CLIENT-SERVEUR



ARCHITECTURE PAIR-À-PAIR

Définition d'un Overlay

Un Overlay est un réseau informatique formant un graphe où les liens sont déterminés avec un critère logique.



Présentation des différents mécanismes importants pour 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 information 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.

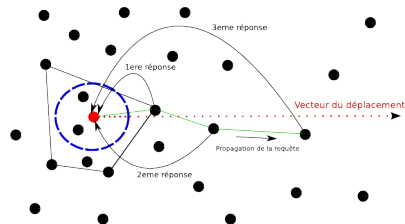


Blue Banana introduit trois états, pour un avatar :

- **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**(ravelling), 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
- Cohérence de la topologie
- Cache Hit et Miss (pour le cache)

En fonction du degré de mobilité.

Le cache des zones denses

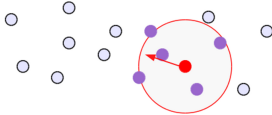
- Explications sur le cache des zones denses
- Les résultats
- Conclusion sur le cache

Comment fonctionne le cache ?

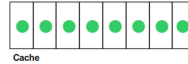
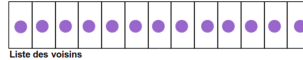
- Chaque nœud de l'environnement a un cache.
- Il est utilisé seulement par les nœuds en état **W**(andering).
- Deux types de cache mis en place (retour simple et retour multiple).
- Sélectionne un nœud en fonction de sa zone de connaissance.



Explications du cache pour les zones denses

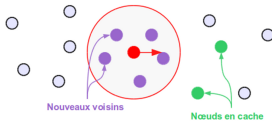


1^{re} phase : le nœud rouge a ses voisins (en violet) et se prépare à se déplacer

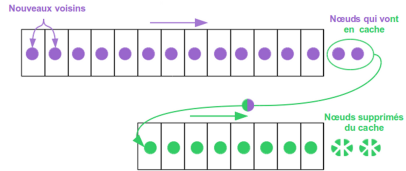




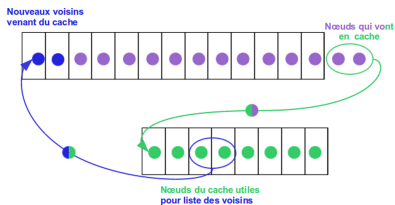
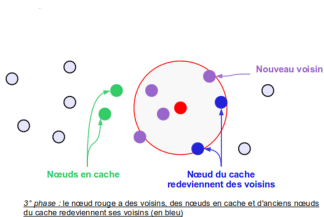
Explications du cache pour les zones denses



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

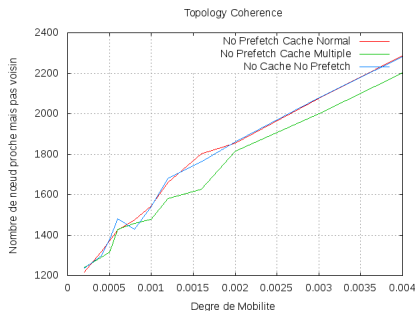
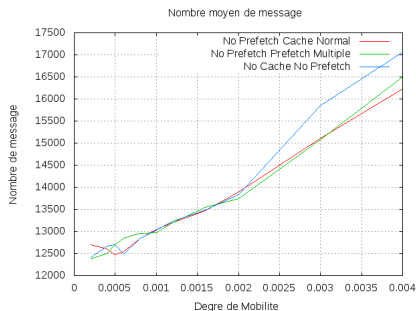


Différents paramètres pour le cache :

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



Résultats pour le cache



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

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.

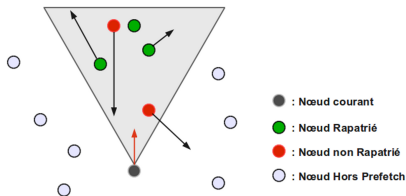
Solution : Choisir plus finement les nœuds qui vont être sélectionnés.

Comment : Regarder la direction des nœuds et leur vitesse.



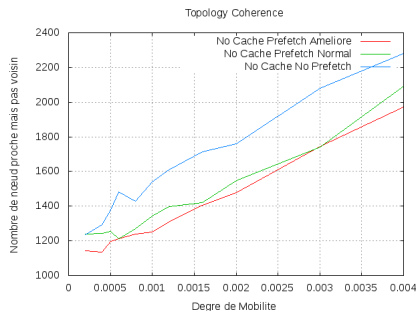
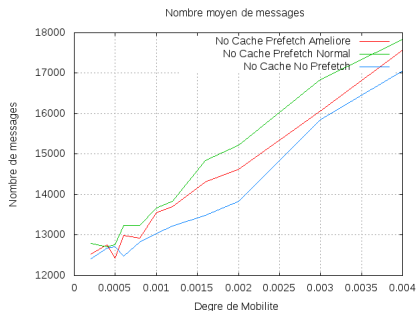
Explications du préchargement amélioré

- ça fait ça
- et puis ça





Résultats pour le préchargement amélioré



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

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

Le préchargement amélioré 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 différentes solutions ont permis d'améliorer la réactivité des réseaux pair à pair pour les MMOGs.
Meilleure cohérence de la topologie et moins de messages que dans Blue Banana.

Merci.

Questions ?