

Distribution dynamique adaptative à l'aide de mécanismes d'intelligence collective

Antoine Dutot

LIH - Université du Havre

24 novembre 2005

Plan

Problème

Simulations
Distribution

Modèle

Formalisation
AntCO²

Expérimentations

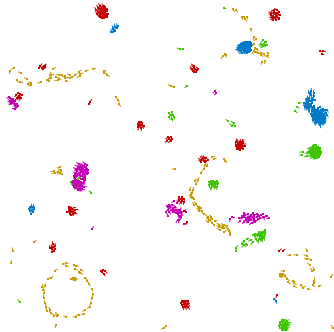
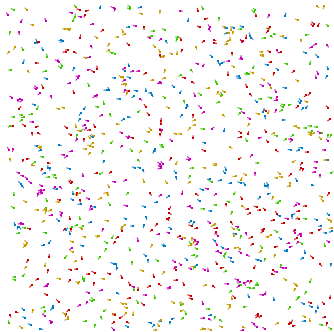
Graphes statiques
Graphes dynamiques

But

- Distribuer une application constituée de multiples entités en interaction de manière dynamique.
- Typiquement adapté à une simulation individu centrée.
- Idées principales :
 - l'application est dynamique et sa distribution doit suivre ;
 - utiliser les organisations au sein de l'application ;

Exemple

Simulation d'écosystème aquatique

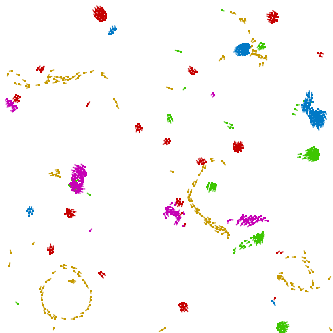


Simulation d'écosystème aquatique

Basée sur les boids de Craig Reynolds (1987).

- Simulation simple ;
- plusieurs espèces ;
- comportement proie/prédateur ;
- grain similaire ;
- organisations émergentes.

Vidéo



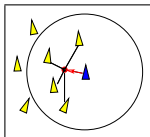
Simulation d'écosystème aquatique

- Nombreuses interactions entre les entités ;
- entités hétérogènes en grand nombre ;
- auto-organisation ;

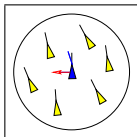
Produisent un
Système complexe.

Simulation d'écosystème aquatique

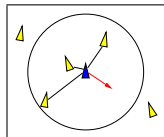
Comportement d'un boid



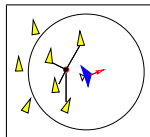
Cohésion



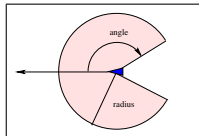
Alignement



Évitement



Répulsion



Vision angulaire

Auto-organisation

Les organisations émergent des interactions entre les boids :

- les règles de cohésion et d'alignement jouent le rôle de feedback positif (morphogénèse) ;
- la répulsion et l'évitement agissent afin de préserver les organisations (morphostase, par feedback positif et négatif).
- les règles de prédation, la vitesse, les obstacles agissent comme feedback négatif qui dispersent les organisations (morphothanatose ?).

Dynamique

Une telle simulation est dynamique à plusieurs niveaux :

- des entités (boids) apparaissent et disparaissent ;
- les interactions changent → les organisations se forment, évoluent, puis se désagrègent ;
- au niveau matériel des ressources de calcul peuvent devenir disponible ou indisponibles.

Distribution

Nous cherchons :

- une distribution qui égalise la charge → répartit les entités de l'application sur plusieurs machines en fonction de leur puissance ;
- minimise les communications → les communications sont dues aux très nombreuses interactions.

Idée

Se servir des organisations pour distribuer.

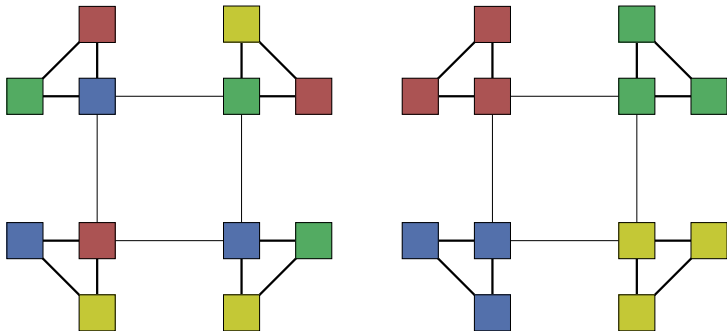
Formalisation du problème

Modéliser l'application sous forme de graphe dynamique

L'application est représentée par un graphe $G = (\mathcal{V}(t), \mathcal{E}(t), \mathcal{C}(t))$ où :

- $\mathcal{C}(t)$ ensemble de p processeurs représentés par p couleurs ;
- $\mathcal{V}(t)$ ensemble de sommets représentant les entités de l'application ;
- $\mathcal{E}(t)$ ensemble d'arcs valués représentant les interactions/communications entre entités.
- les valuations des arcs indiquent l'importance de communications ;
- les colorations indiquent le processeur sur lequel s'exécutent les entités.

Égaliser la charge, minimiser les communications



→ Problème très similaire au partitionnement de graphe, mais sur des graphes dynamiques pondérés.

Problème de la dynamique

Deux approches :

1. Appliquer un algorithme de partitionnement à chaque fois que le graphe change ;
 - Coûteux, le graphe peut avoir changé durant le calcul ;
 - Il faut recommencer chaque étape de calcul du début ;
2. Utiliser une approche *anytime* où :
 - le graphe dynamique est parcouru par de multiples entités ;
 - ces entités stockent la solution directement dans ce graphe ;
 - la solution est donc toujours consultable et mise à jour malgré les reconfigurations continues.

Méthode de distribution

Utilisation d'un algorithme fourmi...

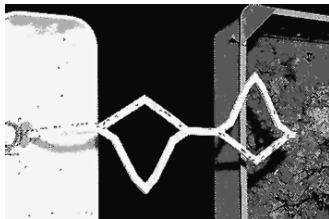


Inspiration : optimisation et fourmis

Dans les années 1980, Deneubourg et son équipe ont montré que les fourmis ont la capacité de trouver le chemin le plus court entre leur nid et une source de nourriture, et ceci de manière adaptative et robuste.

- auto-organisation ;
- adaptativité ;
- robustesse ;

Applet



AntCO² : Principe

Utiliser les organisations : pour cela on cherche à colorer les «clusters» d'entités en forte communication.

- On crée plusieurs colonies de fourmis chacune d'une couleur représentant la ressource de calcul pour laquelle elle travaille ;
- les fourmis visitent le graphe de sommet en sommet suivant un processus itératif ;
- à chaque étape, les fourmis, elles aussi colorées, déposent une petite quantité de phéromones de leur couleur sur les arcs qu'elles traversent ;
- à chaque étape, la phéromone s'évapore.

AntCO²

Coloration

La couleur d'un sommet est déterminée par la phéromone prédominante sur tous ses arcs incidents.

AntCO²

Afin de répartir le graphe en sous-graphes en fonction des organisations très communicantes, les fourmis ont un comportement basé sur :

- la compétition entre les colonies ;
- la collaboration au sein de la colonie ;
- l'attraction par les arcs de forte communication (de poids élevé).

AntCO² : Attraction

- À chaque étape, les fourmis traversent un arc ;
- cet arc est choisi avec un probabilité ;
- cette probabilité est proportionnelle à son importance.

Attraction

Ce comportement favorise la minimisation des communications en faisant en sorte que les fourmis déposent plus de phéromone sur les arcs de forte communication.

AntCO² : Collaboration

- Si une fourmi se trouve dans une zone (un sommet et son voisinage direct) où trop d'autres fourmis de sa couleur se trouvent il y a surpopulation ;
- dans ce cas la fourmi meurt.
- une autre fourmi de la même couleur éclos aussitôt ailleurs dans le graphe.

Collaboration

Ce comportement permet d'égaliser la charge en s'assurant qu'il n'y a pas de zones trop fortement chargée en fourmis et que le nombre de fourmis d'une colonie influe directement sur le nombre d'entités colonisées → les aires du graphe colonisées sont donc proportionnelles aux nombre relatif de fourmis dans les colonies.

AntCO² : Compétition

- Si une fourmi est dans une zone où la phéromone prédominante est d'une autre couleur que la sienne, et que la proportion de sa couleur est en dessous d'un certain seuil, la fourmi est attaquée ;
- dans ce cas elle meurt ;
- une autre fourmi de la même couleur éclos aussitôt ailleurs dans le graphe.

Collaboration

Ce comportement permet la colonisation en détruisant les fourmis des couleurs concurrentes et permet ainsi la création de clusters d'entités colorés de manière uniforme.

Critères de qualité

- *Minimisation des communications :*

$$r_1 = \frac{\text{communications entre deux processeurs distincts}}{\text{toutes les communications}}$$

- *Égalisation de la charge :*

$$r_2 = \frac{\text{taille de la plus petite partition}}{\text{taille de la plus grande partition}}$$

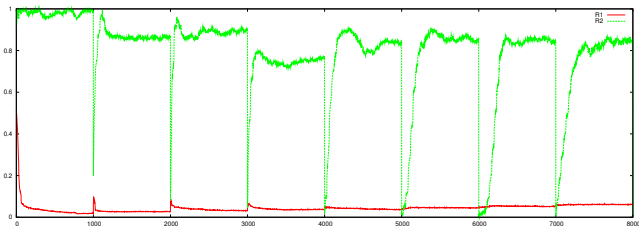
Sur des graphes statiques

- Une grille 30×30 non pondérée.

Avec des graphes statiques

Et des ressources de calcul dynamiques

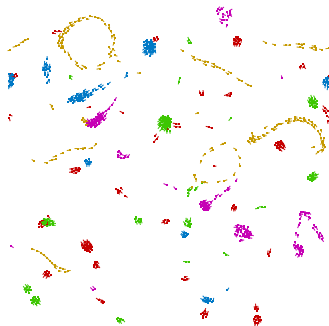
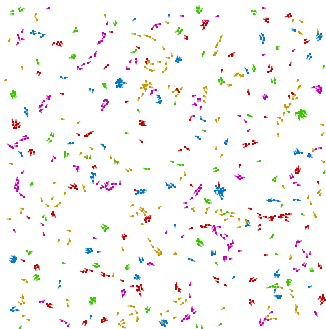
Évolution de r_1 et r_2 avec un graphe de 4000 sommets et 15000 arcs, en ajoutant une colonie et donc un processeur toutes les 2000 étapes.



Avec des graphes dynamiques

Écosystème aquatique

Une simulation simple d'écosystème ou 300 poissons de 4 espèces évoluent.



Merci

Questions ?