

Calcul distribué du jeu de la vie

Travail demandé

Le travail demandé comporte une implantation en RMI, RPC, Corba ou SOAP. Un rapport court (2 à 4 pages) devra accompagner le projet en format papier. Ce rapport devra

- expliquer vos choix de modélisation et lever les ambiguïtés du sujet ;
- les algorithmes principaux sous forme synthétique (pas le code!) ;
- présenter la répartition du travail ;

Implantation

- Votre programme devra utiliser le langage Java ou C (au choix), et devra fonctionner sur la machine *turing*.
- L'interface utilisateur est graphique.
- Une petite explication de l'utilisation de votre programme est aussi nécessaire.

Remise du projet

La remise et l'évaluation du projet se fera la **semaine du 15 Décembre**. Les sources seront envoyées, la même semaine, par courrier électronique à votre responsable, à savoir :

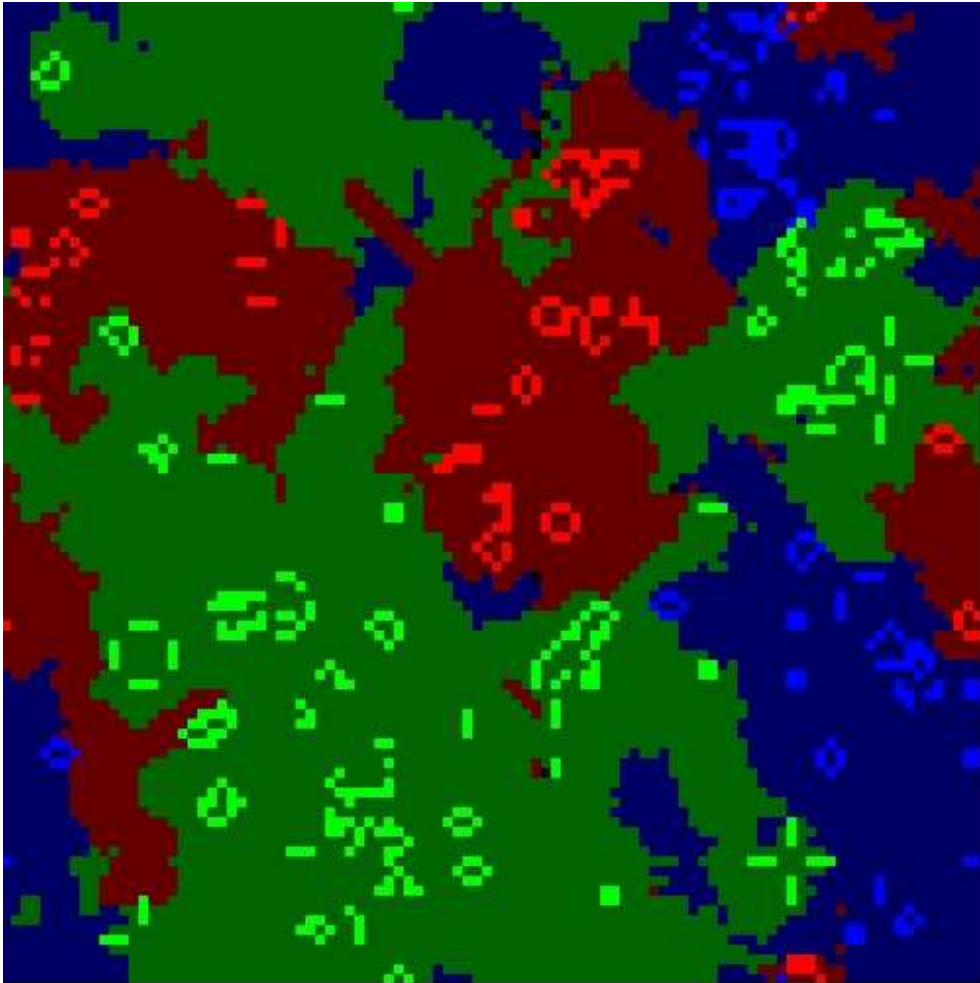
- Benoit Sonntag : benoit.sonntag@lisaac.org

Une soutenance sera organisée la même semaine (A l'heure actuelle, nous avons pas encore fixé de date).

Réalisation du projet

- La réalisation de ce projet devra se faire impérativement **par groupe de deux (ou trois)** pour que vous vous répartissiez le travail au sein du groupe.
- Comme tout cahier des charges, celui-ci ne peut être exhaustif. En cas d'ambiguïté, préciser votre interprétation personnelle, et éventuellement les questions à poser à votre interlocuteur (responsable de projet, futurs utilisateurs, etc.). Toute solution cohérente, justifiée et non contradictoire avec le cahier des charges sera acceptée.

Sujet



L'objectif est de réaliser l'algorithme du jeu de la vie en distribué pour augmenter la vitesse des générations.

Vous trouverez les règles à l'adresse suivante :

http://fr.wikipedia.org/wiki/Jeu_de_la_vie

Le serveur devra réaliser une représentation graphique asynchrone du territoire, pendant que plusieurs clients feront évoluer leur population.

Vous devez évaluer la capacité de chaque machine (potentiellement hétérogène) et profiter au mieux de leurs puissances de calculs. Par exemple, vous pouvez profiter des différents *core* en utilisant des *threads*. Aussi, la répartition et la densité des populations n'étant pas forcément uniforme le simple découpage en rectangle de taille équivalente n'est pas souhaitable.

Le nombre de machine est variable et leurs apparitions au sein du système de calcul est dynamique. C'est à dire, qu'un ordinateur peut demander à tout moment d'intervenir dans le calcul, et donc, la charge doit être rééquilibrée.

Good luck!