

# Simulateur d'automates cellulaires

## Les problématiques

J.-C. Dubacq

Laboratoire d'informatique de Paris-Nord  
Université Paris 13

18 mai 2009

# Plan

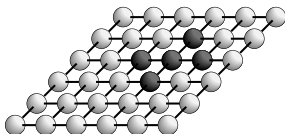
- 1 Les automates cellulaires
  - Définitions et notions de base
  - Un tour d'horizon des automates
- 2 Simulateur d'AC sur grilles
  - Grilles de calcul
  - Les questions soulevées
- 3 Annexe

# Plan

- 1 Les automates cellulaires
  - Définitions et notions de base
  - Un tour d'horizon des automates
- 2 Simulateur d'AC sur grilles
  - Grilles de calcul
  - Les questions soulevées
- 3 Annexe

# Définitions

```
#include <stdca.h>
```



- Élément de base : la configuration
- Configuration : éléments organisés en grille de dimension  $d$  (ligne, plan), tous identiques, prenant un nombre de valeurs finies
- Évolution de configurations grâce à une table de transition *locale* et *uniforme*
- Voisinage : ensemble des cellules dont dépend l'état d'une cellule après une itération de l'automate. Identique pour toutes les cellules
- Notion de temps discret

# Définition formelle

## Définition (Automate cellulaire)

Un automate cellulaire de dimension  $d$  est un triplet  $(\mathcal{S}, V, f)$  où  $\mathcal{S}$  est un ensemble d'états,  $V$  un sous-ensemble fini de  $\mathbf{Z}^d$  et  $f$  une fonction de  $\mathcal{S}^{|V|} \rightarrow \mathcal{S}$ . Une *configuration* est un élément de  $\mathcal{S}^{\mathbf{Z}^d}$ .

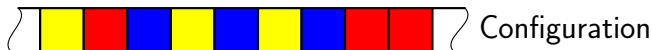
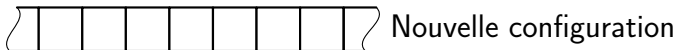


Configuration

# Définition formelle

## Définition (Automate cellulaire)

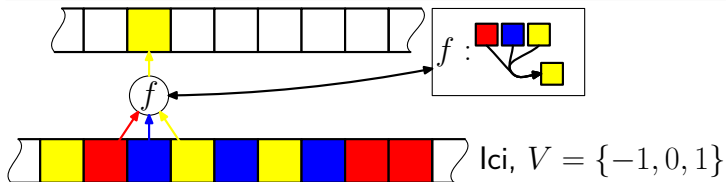
Un automate cellulaire de dimension  $d$  est un triplet  $(\mathcal{S}, V, f)$  où  $\mathcal{S}$  est un ensemble d'états,  $V$  un sous-ensemble fini de  $\mathbf{Z}^d$  et  $f$  une fonction de  $\mathcal{S}^{|V|} \rightarrow \mathcal{S}$ . Une *configuration* est un élément de  $\mathcal{S}^{\mathbf{Z}^d}$ .



# Définition formelle

## Définition (Automate cellulaire)

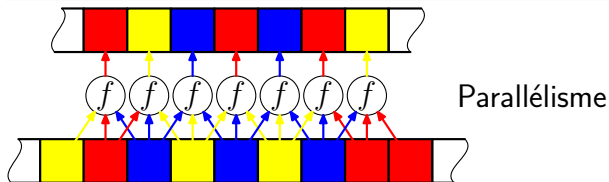
Un automate cellulaire de dimension  $d$  est un triplet  $(\mathcal{S}, V, f)$  où  $\mathcal{S}$  est un ensemble d'états,  $V$  un sous-ensemble fini de  $\mathbf{Z}^d$  et  $f$  une fonction de  $\mathcal{S}^{|V|} \rightarrow \mathcal{S}$ . Une *configuration* est un élément de  $\mathcal{S}^{\mathbf{Z}^d}$ .



# Définition formelle

## Définition (Automate cellulaire)

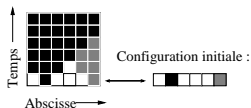
Un automate cellulaire de dimension  $d$  est un triplet  $(\mathcal{S}, V, f)$  où  $\mathcal{S}$  est un ensemble d'états,  $V$  un sous-ensemble fini de  $\mathbf{Z}^d$  et  $f$  une fonction de  $\mathcal{S}^{|V|} \rightarrow \mathcal{S}$ . Une *configuration* est un élément de  $\mathcal{S}^{\mathbf{Z}^d}$ .



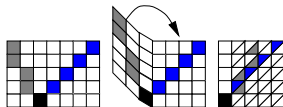
# Diagrammes espace-temps

Du bas vers le haut

- Représentation des données par des diagrammes espace-temps



- Opérations simples sur les diagrammes  $\Rightarrow$  Transformations systématiques sur l'AC

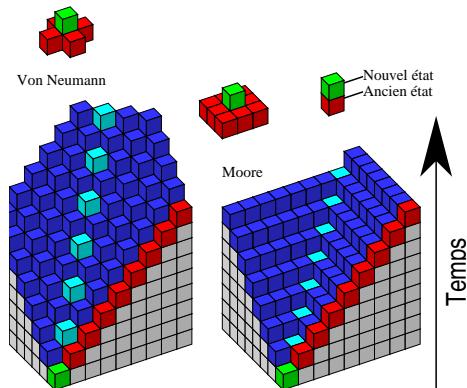


- Exemple du pliage
- Grande composabilité des automates cellulaires : possibilité de programmer de façon locale
- État quiescent : état tel que  $f(q, \dots, q) = q$

# Signaux

- Régularités du diagramme espace-temps  $\implies$  Notion de signal  $\implies$  Nouvelle vision : une cellule reçoit et réemet des signaux
- Émergence d'autres visions des automates cellulaires : moodèle des boules de billard, automates cellulaires partitionnés
- Systèmes multi-agents (calcul à base de fourmis par exemple)
- Problème de ressemblance des états
- Signaux simples (pente rationnelle, forme exponentielle) connus. Savoir-faire existant sur les formes atteignables, mais partiel

# Conventions de représentation en 2D



- Configurations en 2D, donc diagramme espace-temps en 3D
- Problèmes de visualisation complète
- Existence d'états quiescents ?



# Plan

- 1 Les automates cellulaires
  - Définitions et notions de base
  - Un tour d'horizon des automates
- 2 Simulateur d'AC sur grilles
  - Grilles de calcul
  - Les questions soulevées
- 3 Annexe

# Simulation par des calculs d'automates cellulaires

## Les automates comme modèles de calcul

- Modélisation de phénomènes physiques à interaction locale exclusive :
  - Croissance de cristaux
  - Modèle d'Ising
  - Percolation dirigée...
- Modélisation d'interactions biologiques :
  - Diffusion de pigments
  - Courants intra-cardiaques en cas de fibrillation...
- Discrétisation de systèmes continus
- Premier modèle de vie artificielle (von Neumann)

# Simulation du calcul par des automates cellulaires

## Les automates comme modèle de calcul

- Modèles universels : être capable de simuler n'importe quelle instance d'un autre modèle de calcul par les automates cellulaires
  - Circuits électroniques
  - Machine de Turing
  - Automates cellulaires
- Possibilité de faire des simulations préservant ou donnant des propriétés intéressantes :
  - Calcul en temps réel
  - Calcul réversible (dit « à conservation d'énergie »)
- Calcul sur des grilles spécifiques :
  - Voisinage asymétrique ou limité
  - Configurations sur des grilles de dimension  $d$  quelconque
  - Configurations sur des graphes de Cayley... [▶ Exemple](#)

# Plan

- 1 Les automates cellulaires
  - Définitions et notions de base
  - Un tour d'horizon des automates
- 2 **Simulateur d'AC sur grilles**
  - Grilles de calcul
  - Les questions soulevées
- 3 Annexe

# La division du travail

Diviser un calcul en 5 000 éléments qui peuvent être calculés par 5 000 processeurs (on dit *nœud*) ne divise pas le temps par 5 000. Pourquoi ?

**Dépendances de données** Lorsqu'on calcule  $A \leftarrow B + C$  et  $D \leftarrow A + E$ , il faut attendre la fin du calcul de  $A$  pour commencer celui de  $D$ .

**Dépendances d'instruction** Lorsqu'on fait un test, on doit attendre d'avoir fini le test pour savoir où on va, ce qui empêche de préparer le calcul suivant.

# Granularité de la division

On divise donc les tâches de calcul en sous-tâches à plus gros grain. Les principaux écueils :

**Latence d'exécution** Le temps mis à créer une nouvelle tâche s'ajoute au temps de calcul proprement dit.

**Débit de données** Selon les emplacements respectifs, les données ne sont pas accessibles de partout à la même vitesse.

**Latence de données** Lorsque les nœuds sont physiquement distincts, il faut établir des connexions pour transporter les données. Pour des petits volumes de données, ce temps peut devenir prépondérant.

# Pourquoi des grilles ?

Les ressources informatiques sont de plus en plus demandées.

- Grilles de calcul
- Grilles de données, réseaux pairs-à-pairs
- Calcul mobile, calcul ubiquitaire
- Communication automatiques inter-nœuds, grilles instantanées !

Qu'est-ce qui ne passe pas à l'échelle ?

- Tolérance aux pannes (nœuds, réseau)
- Circulation de données (pare-feux, débit, quantité)
- Gestion des nœuds intermittents (grilles légères ou *desktop grids*)

# Pourquoi des grilles ?

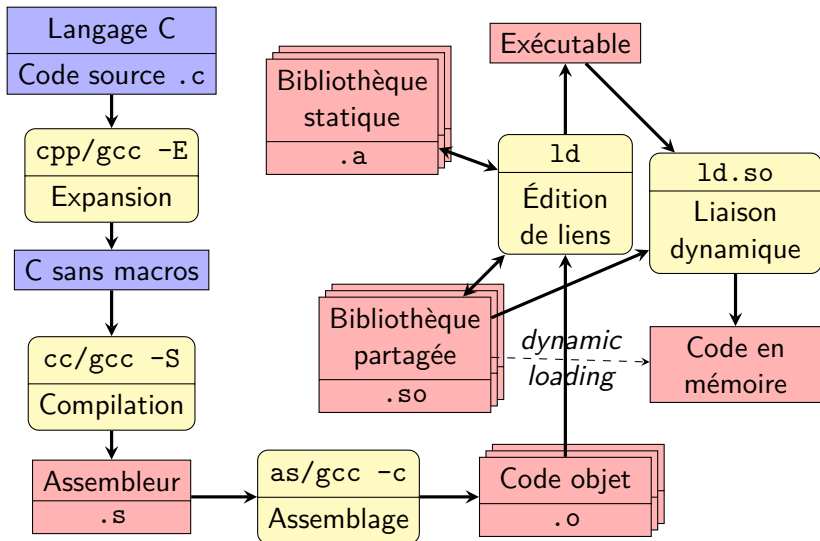
Les ressources informatiques sont de plus en plus demandées.

- Grilles de calcul
- Grilles de données, réseaux pairs-à-pairs
- Calcul mobile, calcul ubiquitaire
- Communication automatiques inter-nœuds, grilles instantanées !

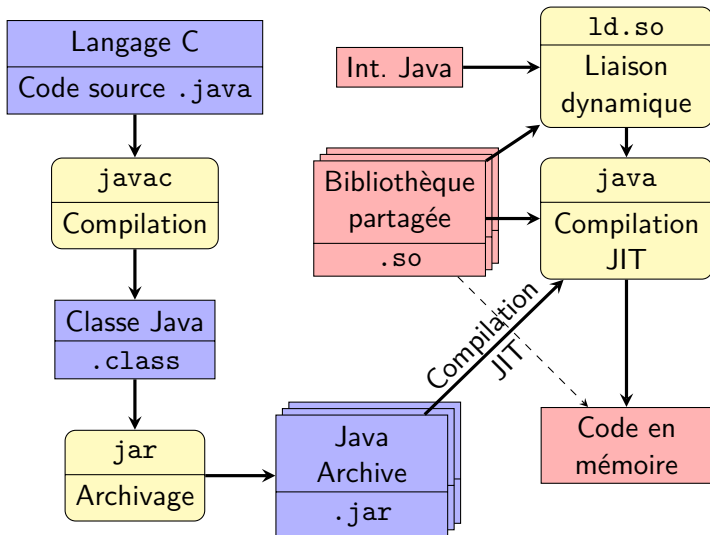
Qu'est-ce qui ne passe pas à l'échelle ?

- Tolérance aux pannes (nœuds, réseau)
- Circulation de données (pare-feux, débit, quantité)
- Gestion des nœuds intermittents (grilles légères ou *desktop grids*)

# Préparer l'exécution sur une grille



# Le cas des langages à bytecode (Java)



# Plan

- 1 Les automates cellulaires
  - Définitions et notions de base
  - Un tour d'horizon des automates
- 2 **Simulateur d'AC sur grilles**
  - Grilles de calcul
  - Les questions soulevées
- 3 Annexe

# L'historique

- Il existe beaucoup de programmes pour des cas particuliers :
  - 1D : Classification de Wolfram
  - 2D : Life, Langton, etc.
  - Souvent en Java ou en C/C++ et assembleur.
- Il existe beaucoup de programmes génériques à visualisation directe
- Il existe des programmes à entrée complètement générique insuffisants pour du calcul massif
- Existence brève d'OpenCA (93–99) pour des machines parallèles à passage de message
- Décomposition du programme en trois parties : entrée, calcul, visualisation

# L'entrée des données

- Fonctions de transition données par différents moyens :
  - Tables de transition (petits CA/PCA/TCA)
  - Fonctions de calcul
  - Signaux
  - Systèmes multi-agents
  - Choix du langage (C, C++, Fortran, assembleur)
- Compilation à la volée ou chargement dynamique
- Génération de configurations :
  - Taille et topologie (tore, bloquée, finie, répétée)
  - Aléatoire initial, impulsion, entrée manuelle, jeux d'expériences

# Le calcul

## Sur les nœuds

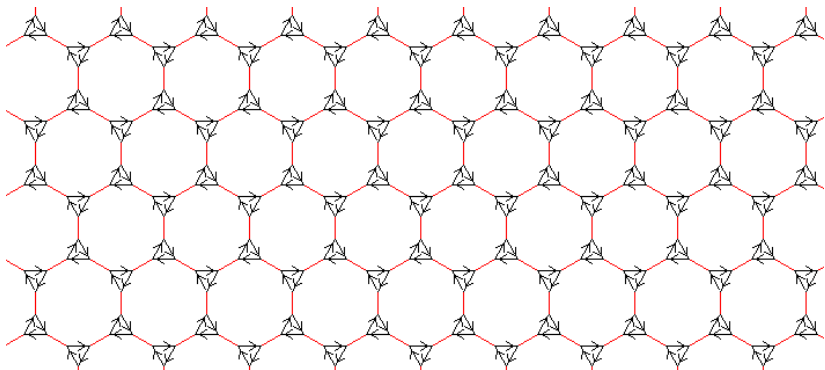
- Gestion optimisée de la mémoire (pointeurs coulissants, par exemple)
- Utilisation d'assembleur, prédictions de branchement
- Gestion de l'hétérogénéité de chaque zone

## Sur la grille

- Évaluation du ratio calcul/transfert de données/recouvrement de calcul
- Gestion de l'hétérogénéité réseau et nœud
- Tolérance aux fautes (disparition de nœuds, pannes)

# La visualisation des données

- Visualisation totale : volume de données très important
- Le choix d'intervalle de sauvegardes peut effacer/masquer des comportements dynamiques importants
- Évaluateur de visualisation partielle pour vision en temps réel ou détection de mouvements intéressants
- État quiescent ou pas ? Dans le cas de 2D (diagramme espace-temps) ou 3D (configuration), la différence est fondamentale
- Collectes de statistiques sur calculs nombreux



► Retour