

---

# Techniques et outils pour la construction massivement parallèle de systèmes de transitions

---

Christophe Joubert

DEA Informatique Systèmes et Communications

19 Juin 2002

*Projet VASY*

*INRIA*



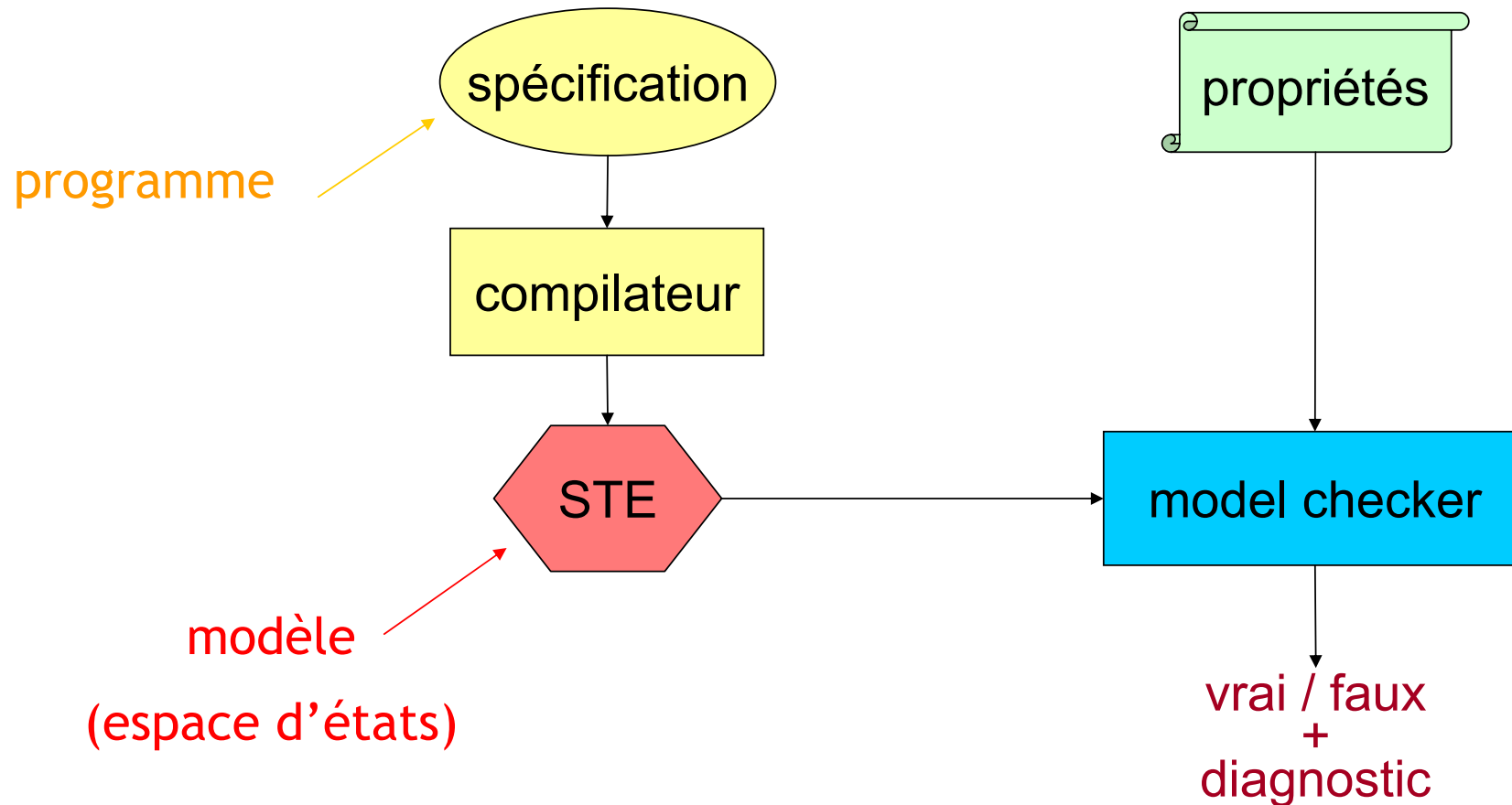
---

# Contexte du travail

---

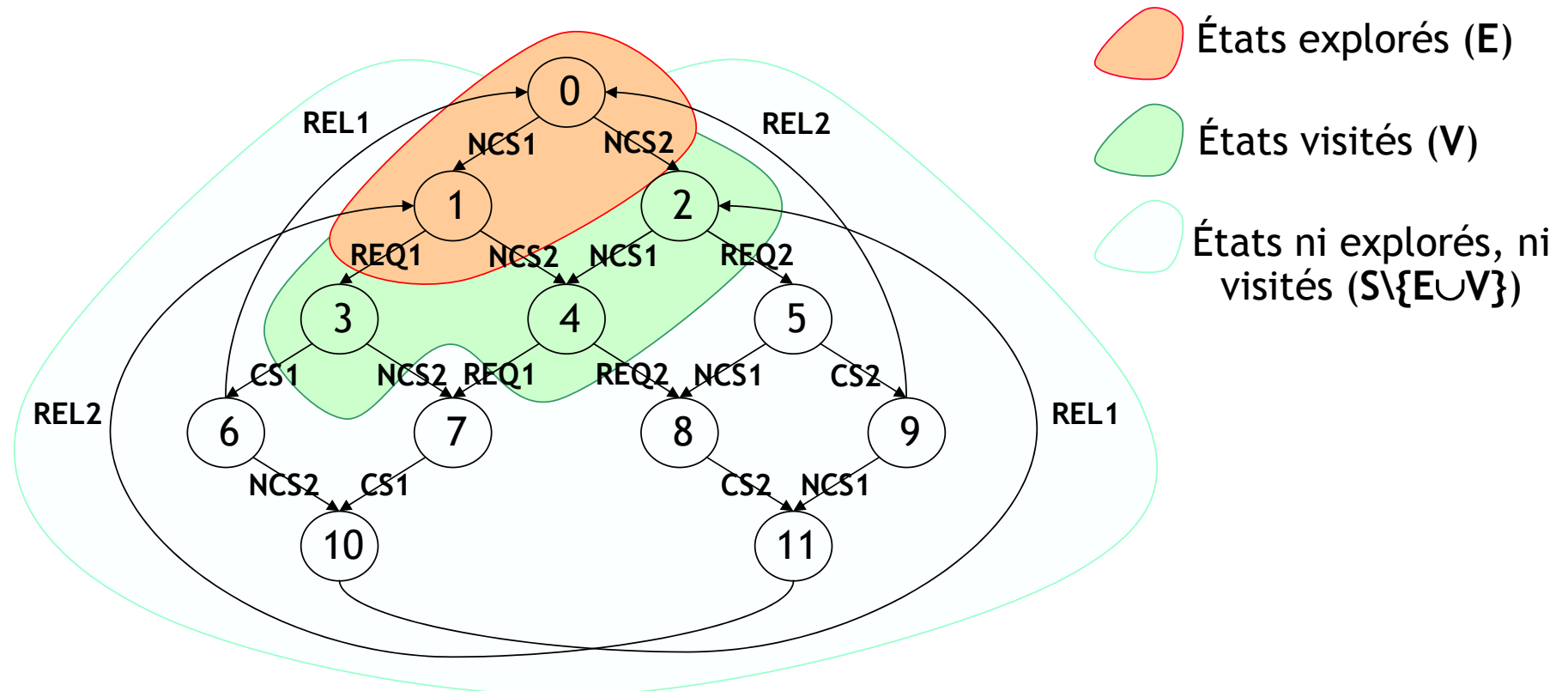
- **Projet VASY** (VAlidation de SYstèmes)
  - ✦ Méthodes formelles appliquées aux systèmes critiques
- **CADP** (Caesar/Aldebaran Development Package)
  - ✦ Boîte à outils pour la vérification de protocoles et de systèmes distribués développée depuis 1986

# La vérification par modèles (model-checking)



# Génération séquentielle d'espaces d'états

- Espace d'états (STE) :  $M = \langle S, A, T, s_0 \rangle$
- Exemple du protocole d'exclusion mutuelle



# Le problème de l'explosion d'états

- Croissance exponentielle du nombre d'états
  - ✦ nombre de processus parallèles
  - ✦ taille des domaines des variables
- Deux solutions complémentaires :
  - ✦ Éviter l'explosion
    - approche symbolique
    - réductions modulo des propriétés
    - abstractions
  - ✦ Pallier l'explosion
    - vérification à la volée
    - compression des états
    - parallélisation de la vérification

---

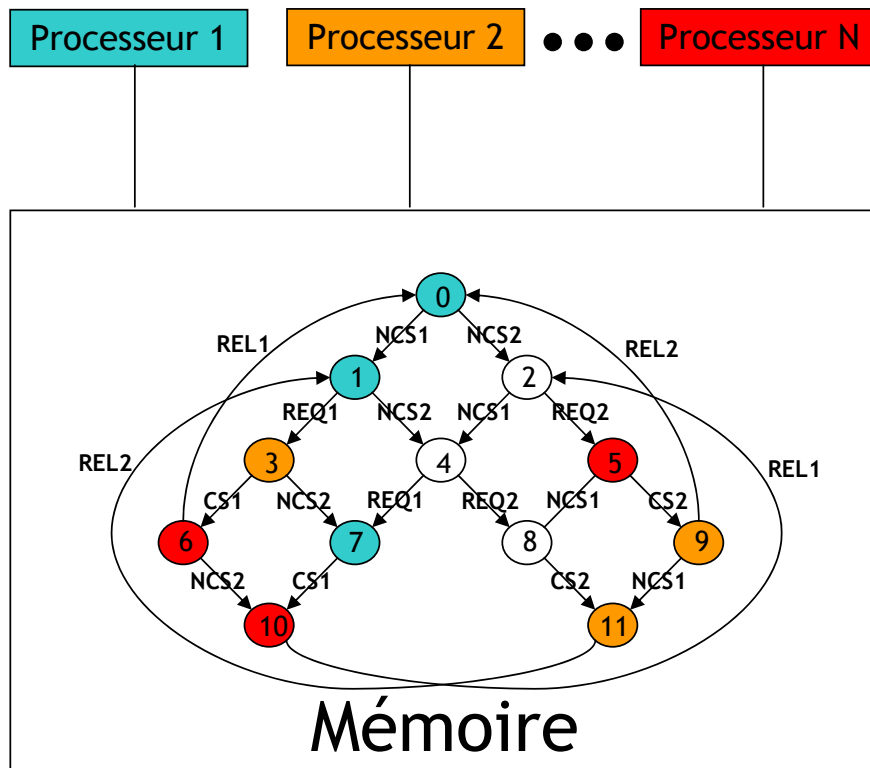
# Objectifs et plan

---

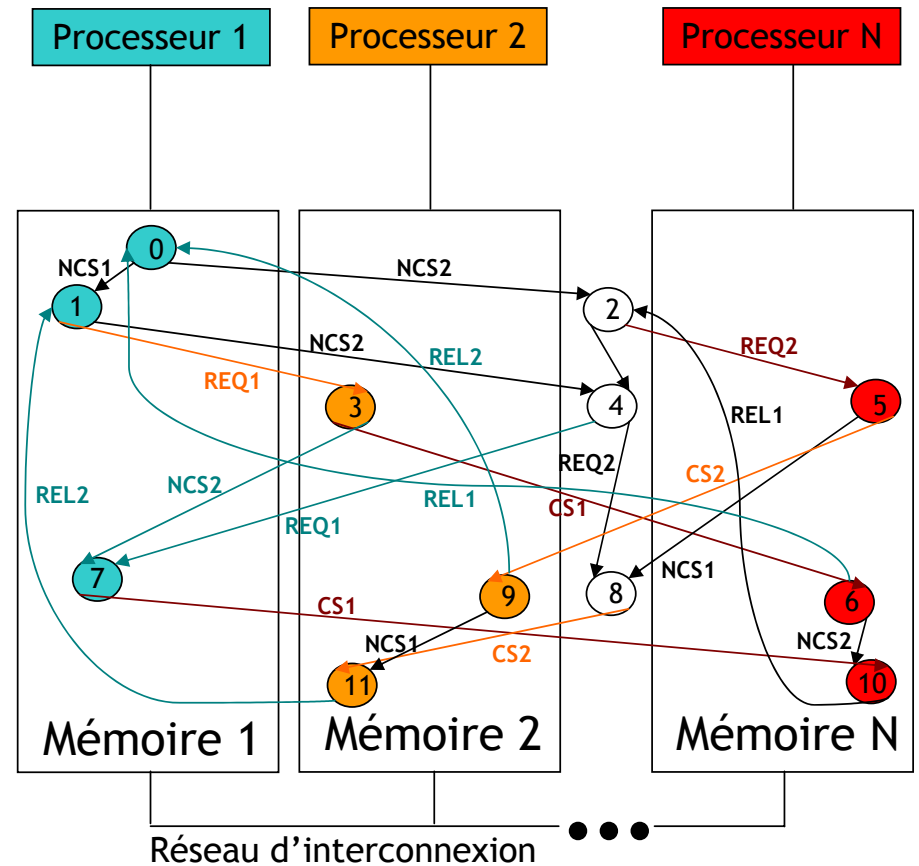
- Génération parallèle d'espaces d'états
  - ✦ État de l'art
- Algorithmes DISTRIBUTOR et COORDINATOR
  - ✦ Gestion distribuée des données dynamiques
- Spécification et vérification formelle
  - ✦ LOTOS et CADP
- Réalisation et expérimentation
  - ✦ Grappe de PC de l'équipe APACHE (ID/INRIA)

# Génération **parallèle** d'espaces d'états

- Architecture parallèle support



Mémoire partagée [Allmaier-97]



Mémoire distribuée [Caselli-95]

# Les composants de la génération parallèle

- Type de vérification parallèle [Inggs-00, Heyman-00]
  - ✦ énumérative (STE explicite), symbolique (BDD), probabiliste
- Langage d'entrée [Stern-Dill-97]
  - ✦ Multi-langages (Open/Caesar), LOTOS, Murphi, Promela, réseau de Pétri
- Format de description du modèle [Nicol-97]
  - ✦ STE (BCG), chaînes de markov, etc.
- Répartition des tâches [Ciardo-01, Allmaier-Horton-97]
  - ✦ hachage statique prédéfini, rééquilibrage dynamique

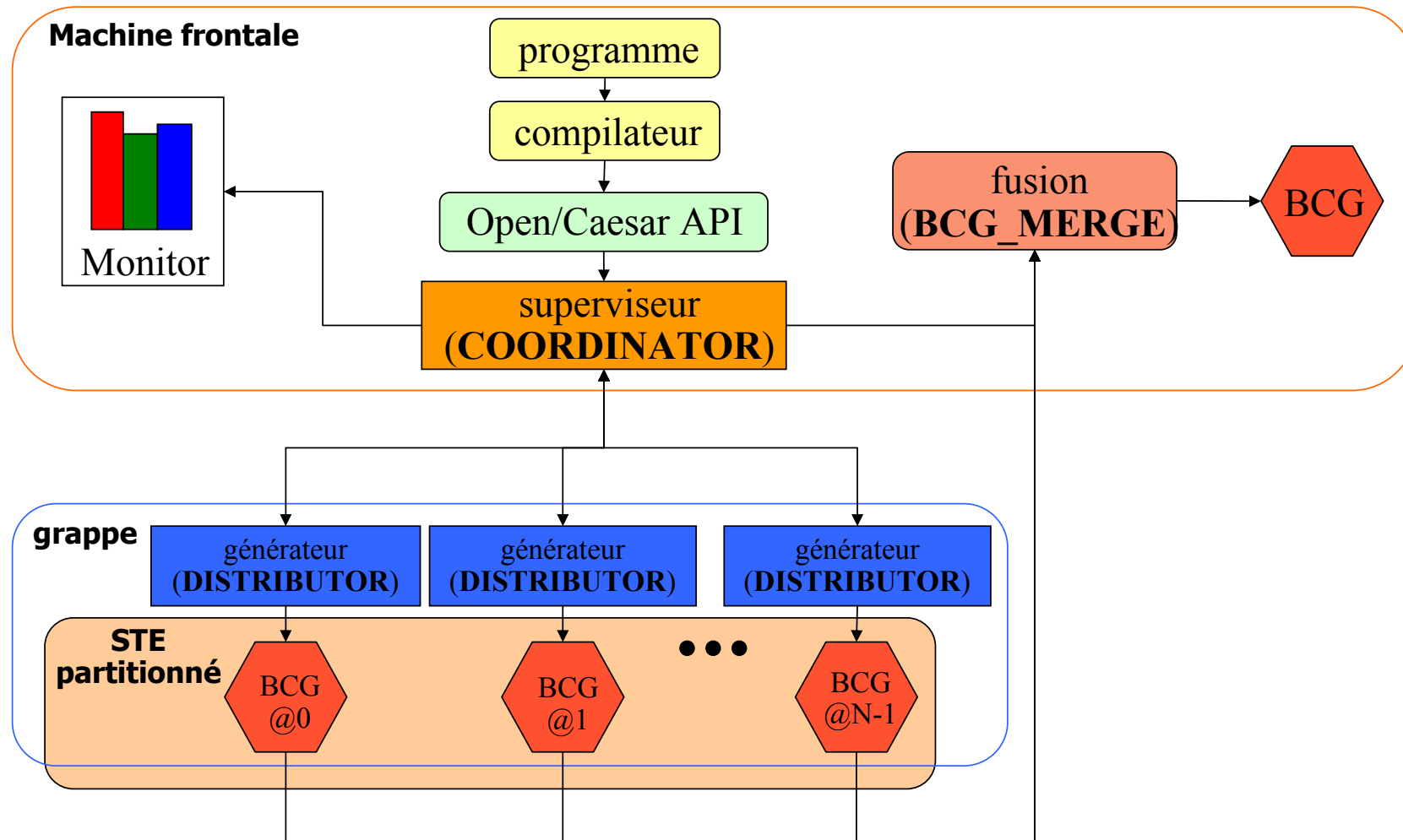


# Les composants de la génération parallèle

- Stockage des états [Allmaier-Kreische-97, Haverkort-97]
  - ✦ **table de hachage**, pile, file, arbre de recherche équilibré, BDDs, etc.
- Type de communication [Ciardo-98, Lerda-Sisto-99]
  - ✦ **Ethernet**, Myrinet, etc., MPI, PVM, **sockets TCP/IP** ou UDP/IP
- Détection distribuée de la terminaison [Matocha-97, Mattern-87]
  - ✦ **sites inactifs**, **plus de messages en transit**, « **four counter method** »
- Performances globales [Knottenbelt-98]
  - ✦ **accélération proche du linéaire**, taille supérieure à  **$10^7$  états**

# Algorithmes **DISTRIBUTOR** et **COORDINATOR**

- Schéma général de la génération distribuée



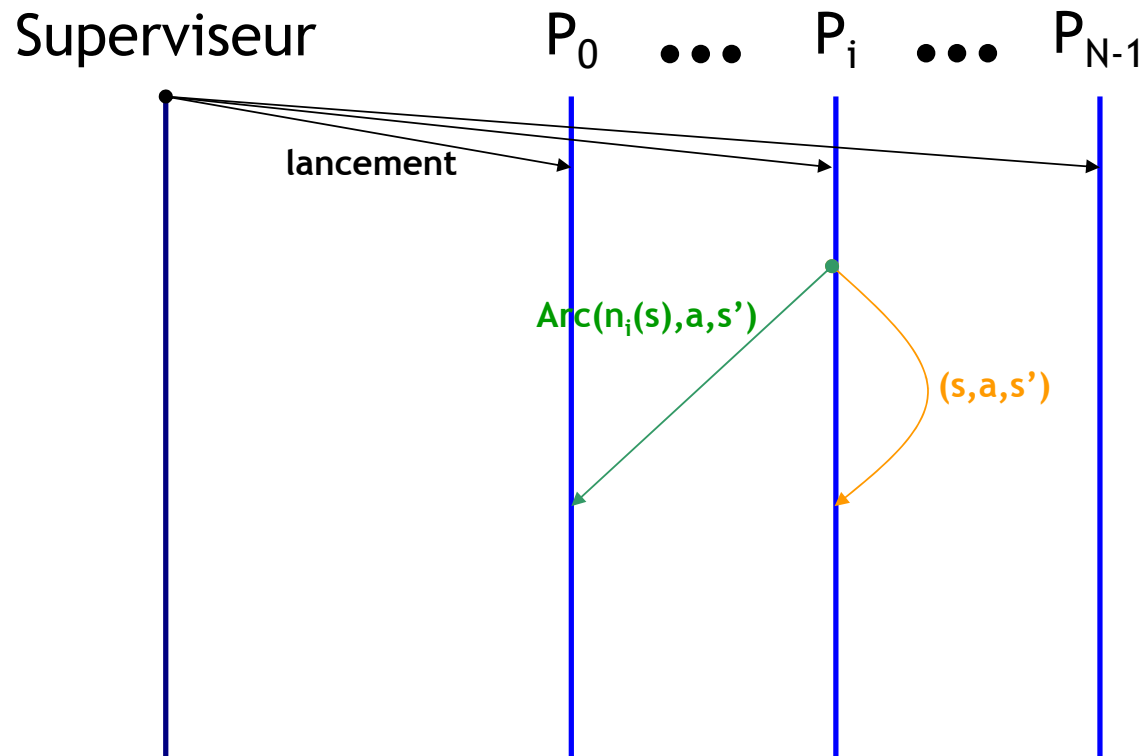
# Cinq protocoles combinés

- DISTRIBUTOR
  - ✦ Protocole de **génération** parallèle
  - ✦ Protocole de détection distribuée de la **terminaison**
- DISTRIBUTOR + COORDINATOR
  - ✦ Protocole de gestion des types de **données dynamiques**
  - ✦ Protocole de détection de **pannes** franches ou d'arrêts utilisateur
  - ✦ Protocole de **visualisation** de l'avancement de la génération
- 5 protocoles entrelacés → complexité algorithmique

# Complexité des protocoles

- DISTRIBUTOR (Générateur)
  - ✦ 4 pages d'algorithme (20 macro états, 24 variables d'état)
- COORDINATOR (Superviseur)
  - ✦ 2 pages d'algorithme (4 macro états, 8 transitions, 9 variables d'état)
- Types de messages : Arc(n,l,s) (*génération*), Rec(k), Snd(k) et End (*terminaison*), Lab(a) (*données dynamiques*), Emg (*pannes ou arrêts utilisateur*), Stats(d), Trm, Init, Stop(d) et Visit(d) (*visualisation*)

# Protocole de **génération** parallèle



Graphe complet ( $N^2$ ) :

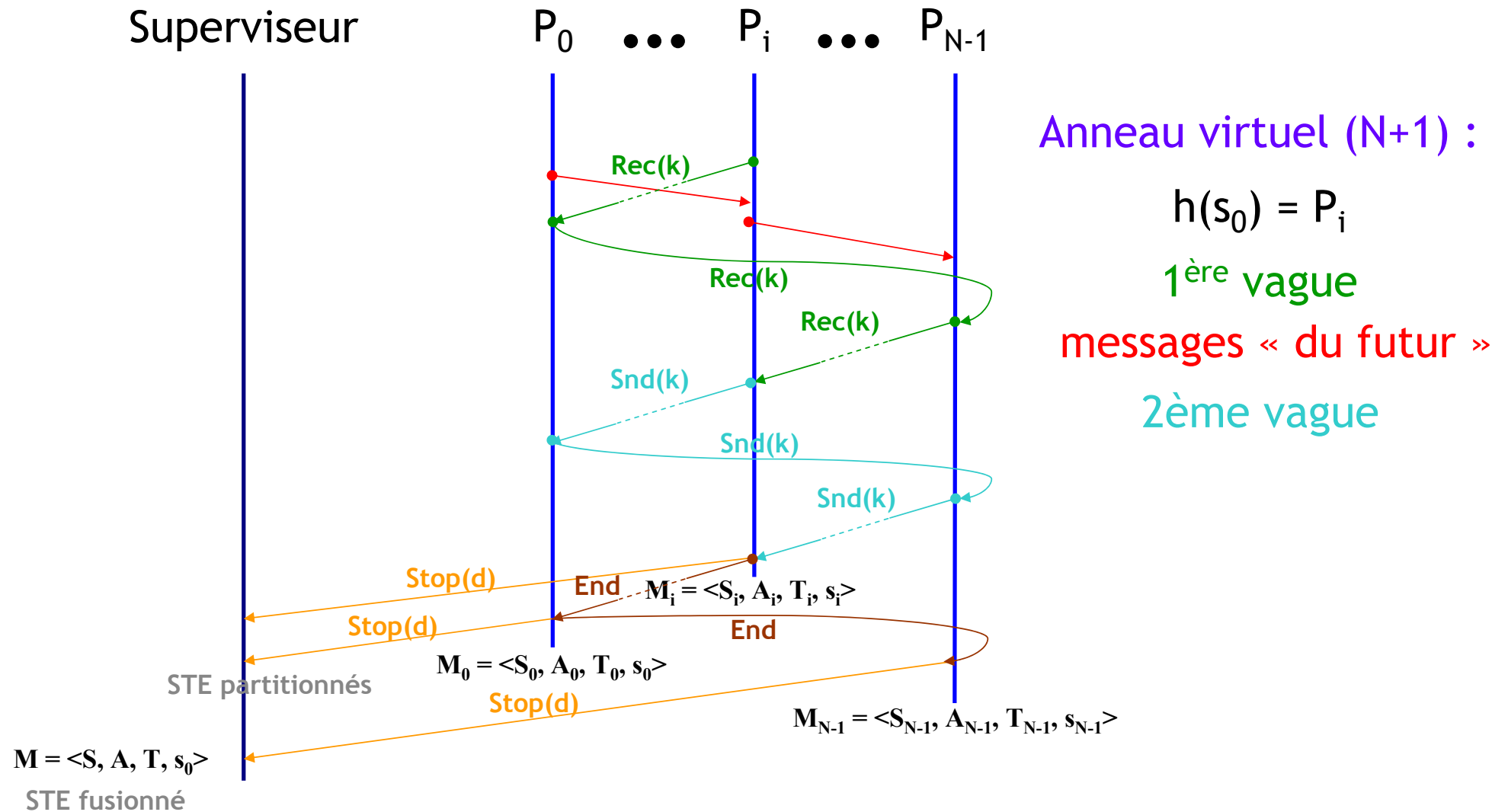
$$h(s_0) = P_i$$

$$(s, a, s') \in \text{succ}(s)$$

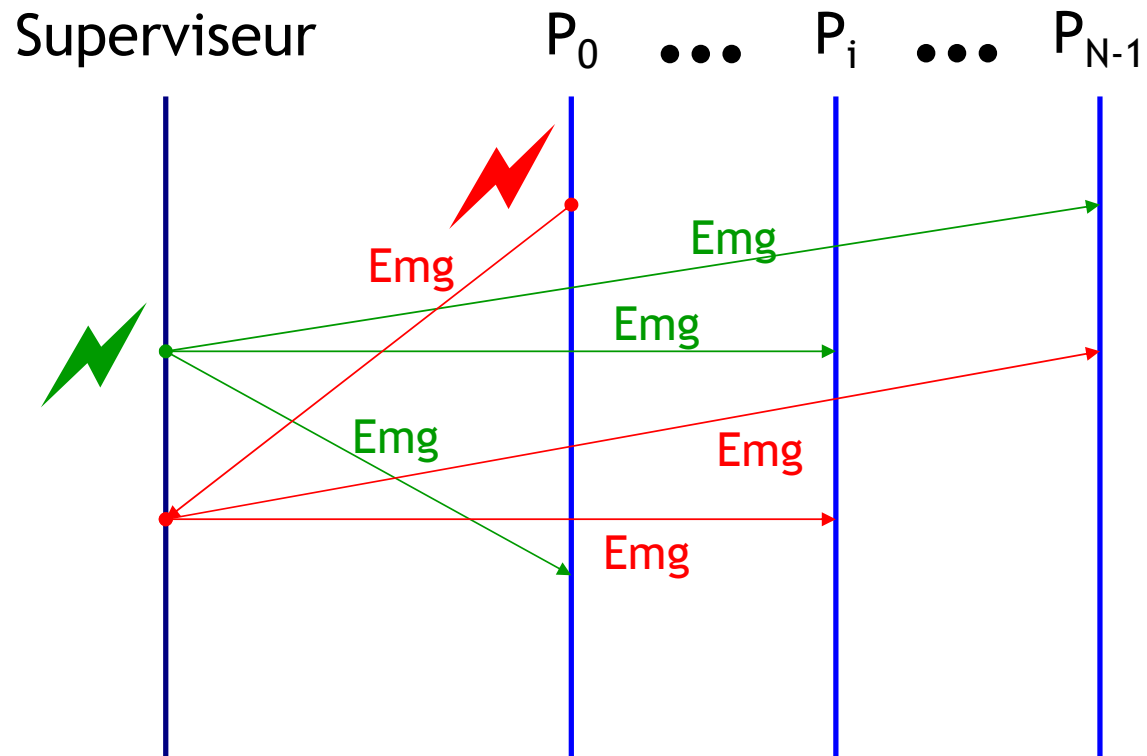
- $h(s') = P_i$

- $h(s') \neq P_i$

# Protocole de détection de la terminaison



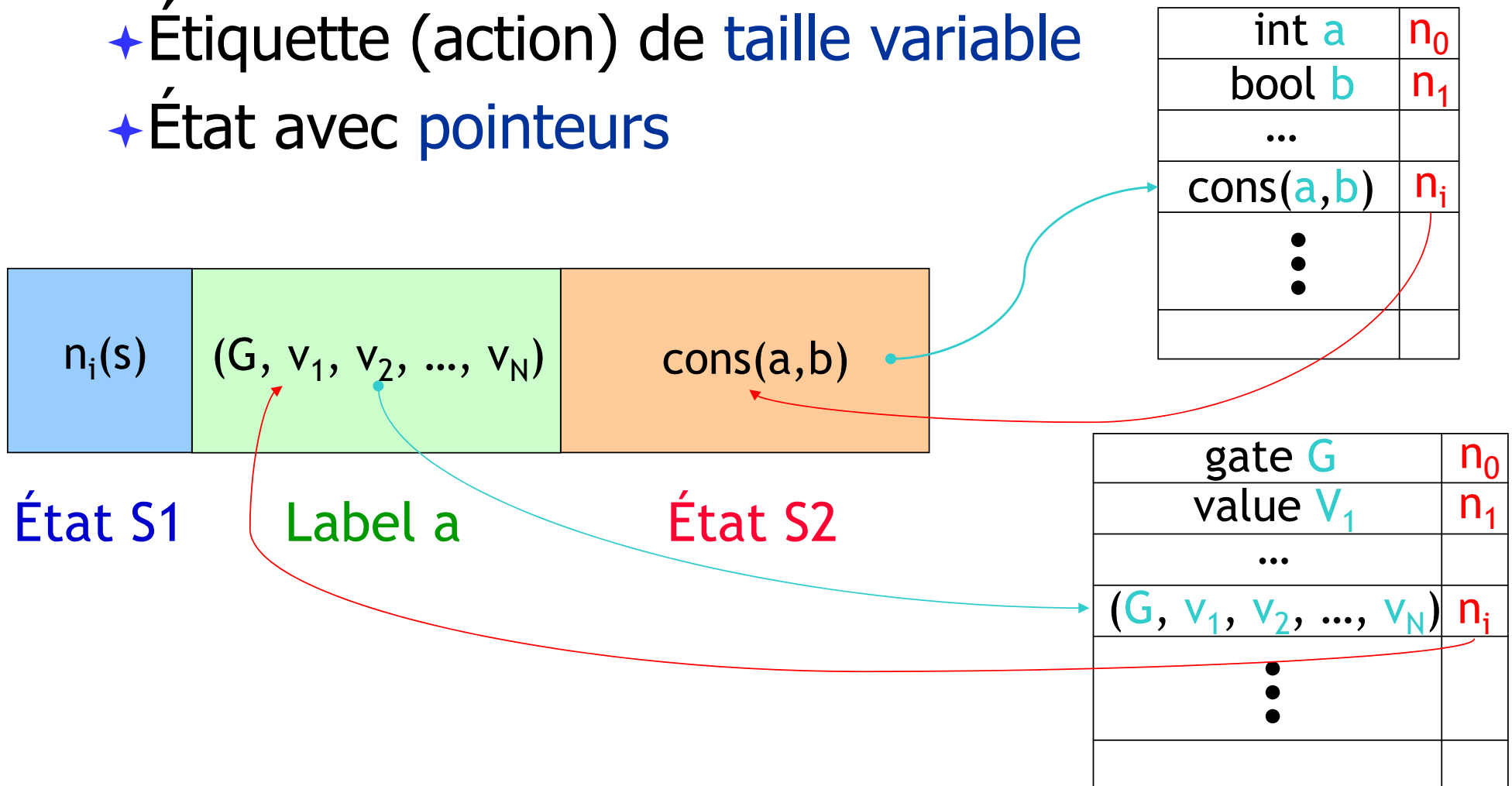
# Protocole de gestion des pannes



Étoile (N) :  
Arrêt de l'utilisateur  
Panne franche

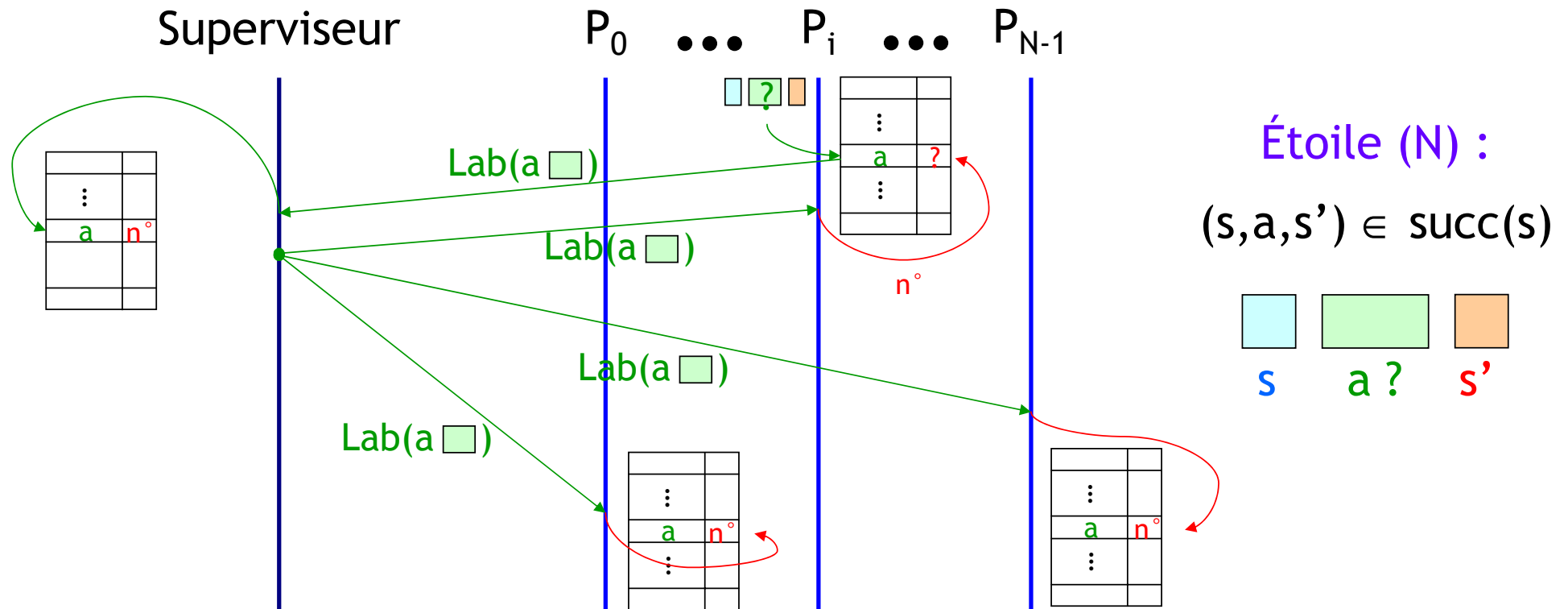
# Problème de la cohérence des termes

- Deux types de données dynamiques
  - ✦ Étiquette (action) de **taille variable**
  - ✦ État avec **pointeurs**



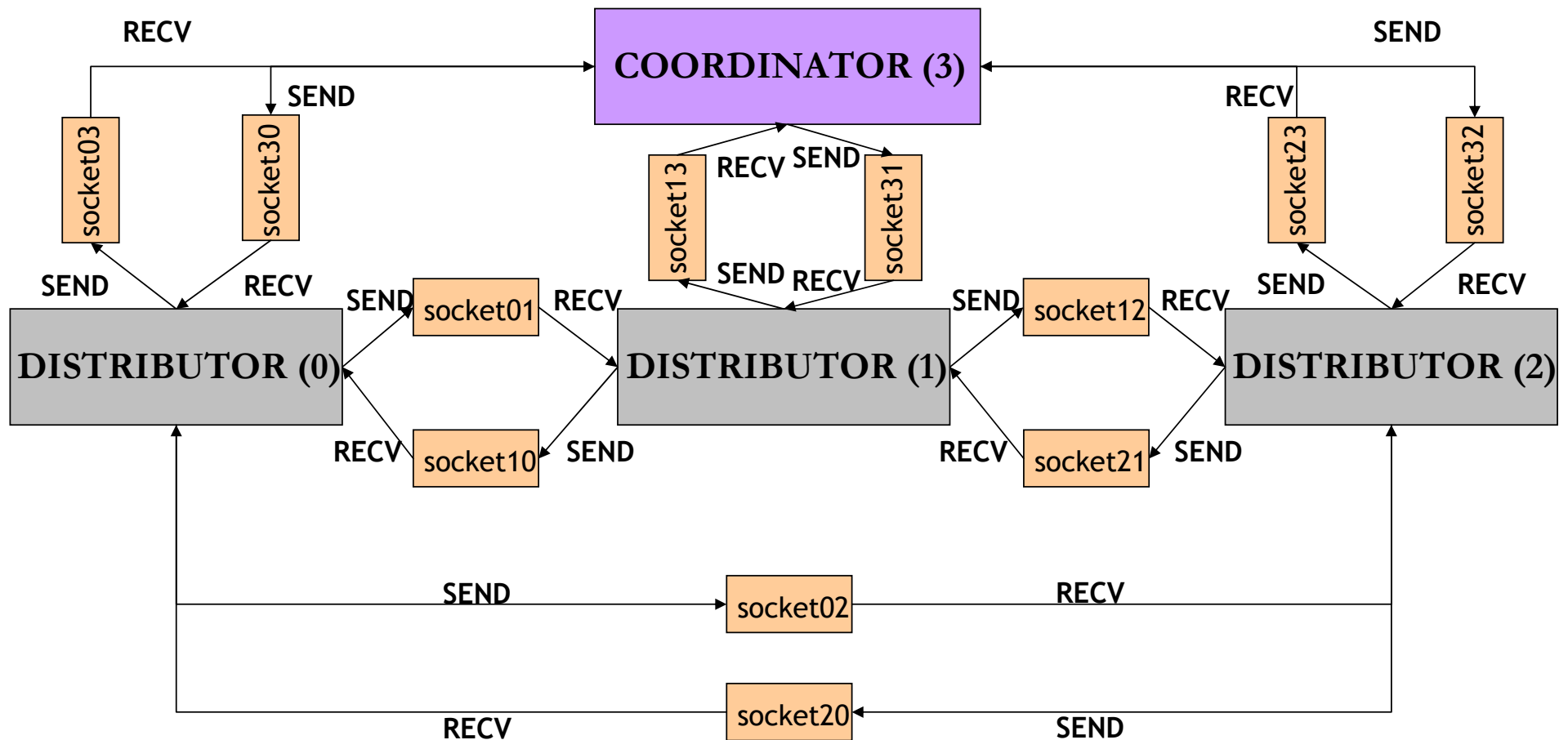


# Protocole de cohérence des termes



# Spécification formelle en LOTOS

- Architecture de la spécification :



# Deux variantes de la spécification

- Spécification 1 (S1) :
  - ✦ Sans processus superviseur (DISTRIBUTOR)
  - ✦ Protocole de **génération** parallèle et de **terminaison** distribuée
- Spécification 2 (S2) :
  - ✦ Avec processus superviseur (DISTRIBUTOR + COORDINATOR)
  - ✦ Les deux protocoles précédents + protocole de **cohérence des étiquettes**
- Paramètre d'entrée : STE du protocole d'exclusion mutuelle par sémaphore

# Vérification avec CADP

- **Logique temporelle** : 5 propriétés vérifiées sur  $S_1$

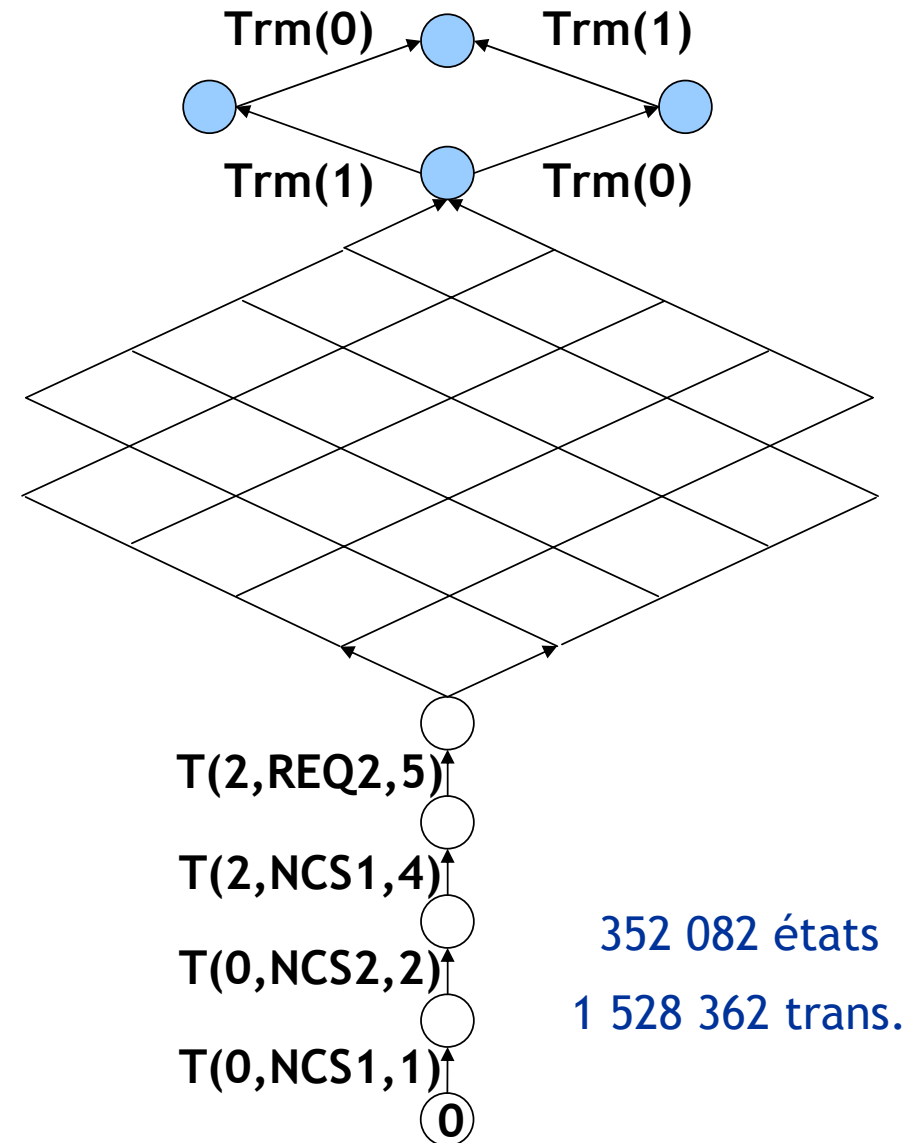
- ✦ **Sûreté** :

- Toutes séquences d'exécution de transitions dans le graphe d'états sont finies
- Sur chaque séquence de transitions menant à un état de blocage, chacune des transitions du STE à générer, est imprimée
- Chaque état de blocage est précédé par une séquence d'actions de terminaison, exactement une par processus dans le réseau
- Aucune action n'est possible après que toutes les actions de terminaison aient été exécutées

- ✦ **Vivacité** :

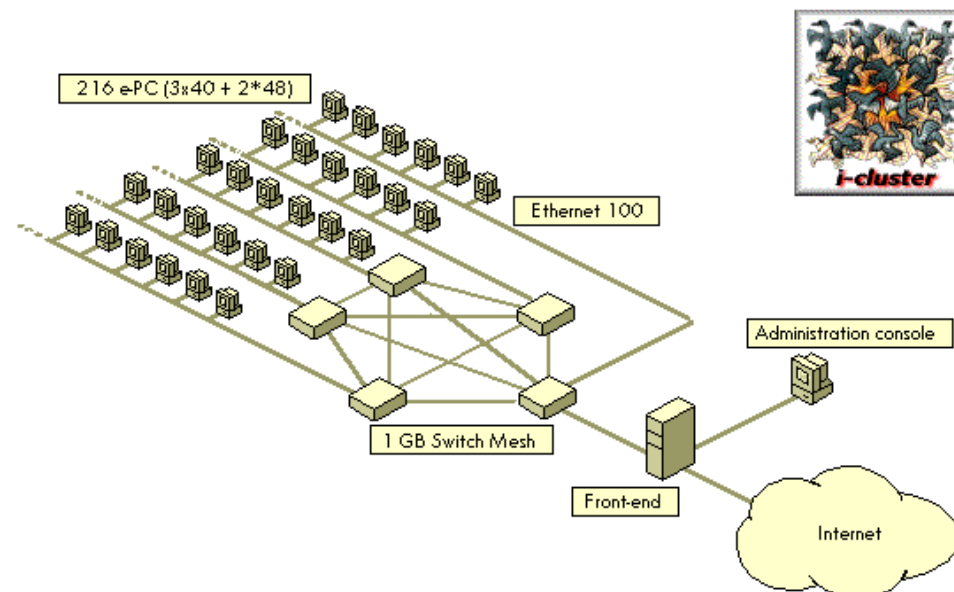
- Un état de blocage est finalement toujours atteint

- **Équivalence** de  $S_1$  et  $S_2$  modulo la bisimulation de branchement  $\rightarrow S_1 \approx S_2$

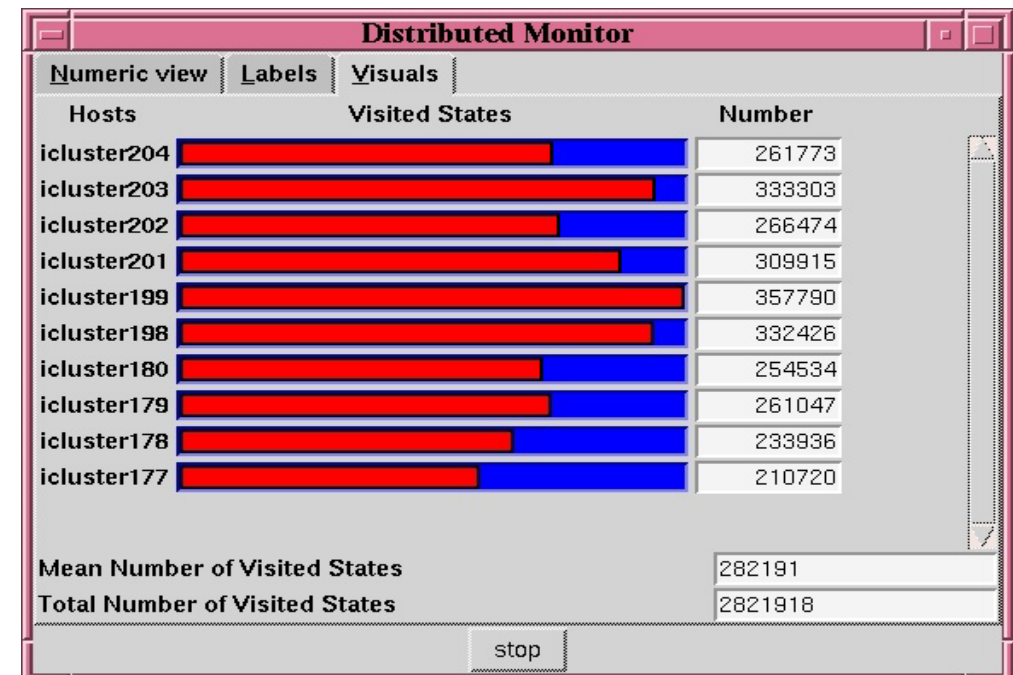
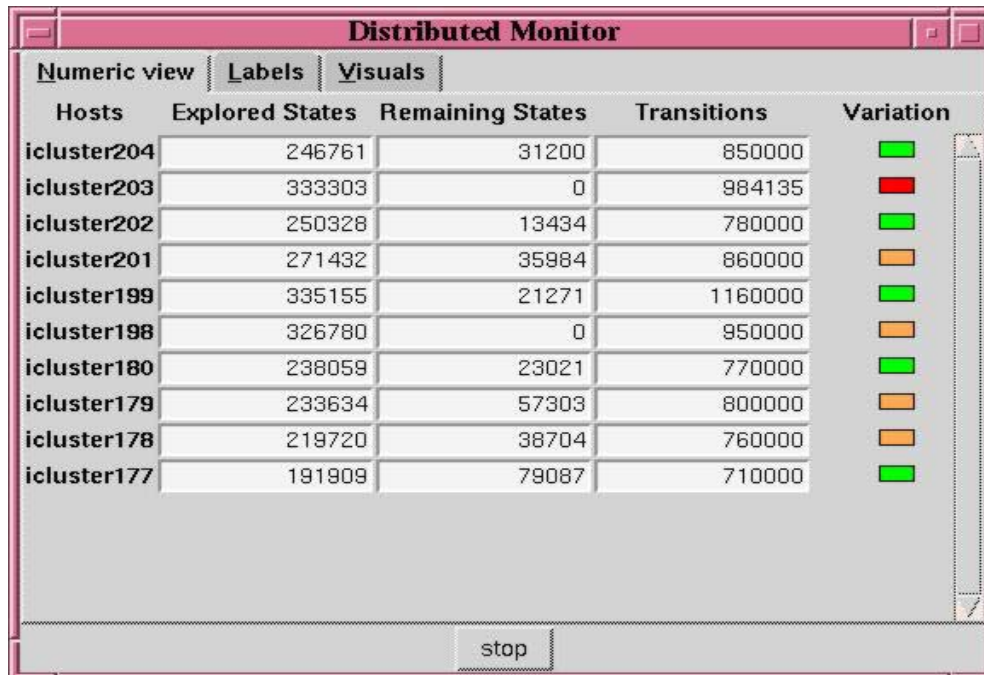


# Réalisation et expérimentation

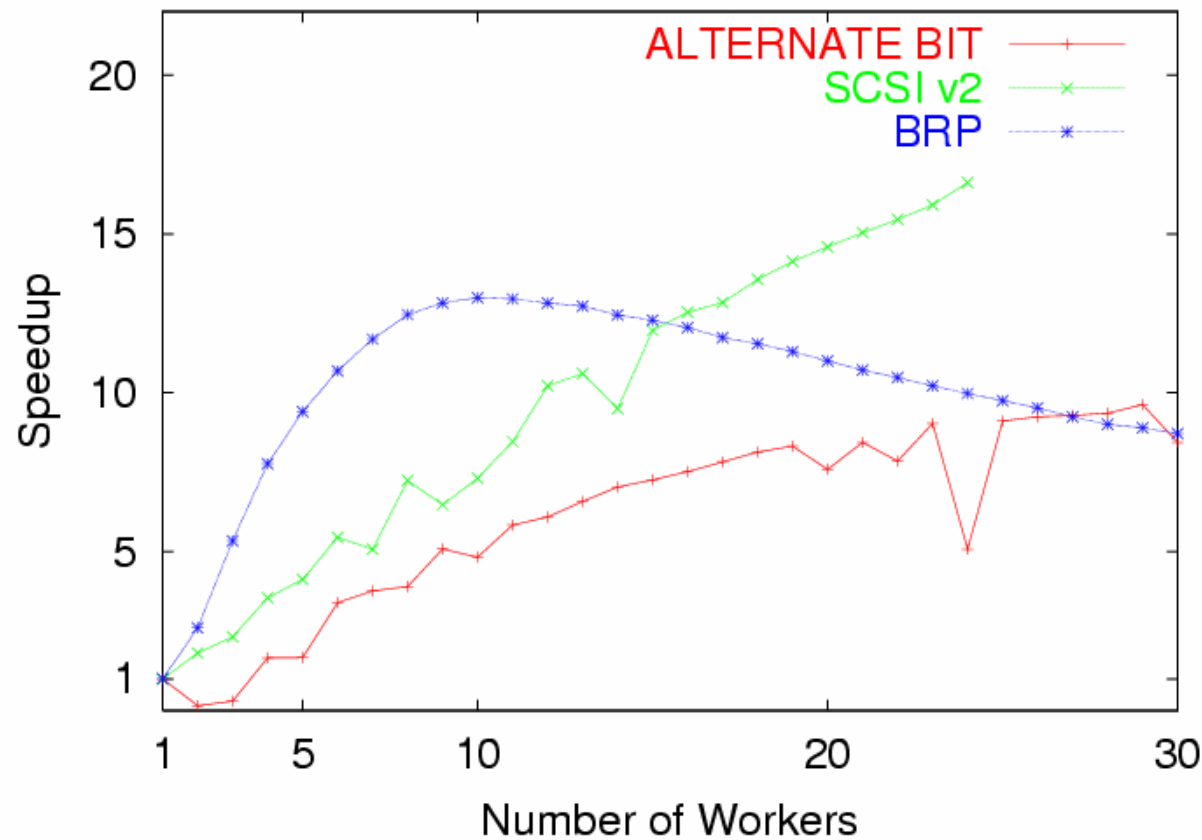
- Intégrations
  - ★ reprise et mise au point d'un prototype existant (Smarandache-Curic, 6000 lignes de C) :
    - taille des labels fixée arbitrairement, bufferisation impropre des messages, ordonnancement non justifié des activités de calculs et de communication, etc.
  - ★ nouveaux algorithmes pour DISTRIBUTOR et COORDINATOR
  - ★ primitives de communication génériques (1000 lignes de C)
- Expérimentations sur la grappe de PC i-cluster



# Visualisation de la progression du système



# Accélération en temps de calcul



# Bilan

- État de l'art exhaustif
  - 20 articles comparés selon 40 critères = 800 questions à répondre
- Approche générale
  - indépendants du langage de spécification
  - utilisant des architectures massivement parallèles répandues
- Problèmes nouveaux et utiles
  - gestion de la terminaison, des types dynamiques, des étiquettes de taille variable, dans la génération répartie de STE
- Protocole de génération distribuée
  - mono-thread avec priorité entre les activités
  - anneau virtuel et étoile
  - étude par évaluation de la complexité (nombre de messages)
  - premières expérimentations de vérification par model-checking



# Perspectives

- Implémentation
  - Finir l'intégration du protocole de cohérence des termes
  - Séparation sur deux threads des calculs et des communications
- Expérimentation
  - Nouvelles versions de la table d'états et du format BCG (>16M états)
  - Expériences et études de performances avec la grappe stable
- Preuve formelle de correction
  - Vérification indépendante du STE à générer
- Parallélisation de la vérification énumérative
  - Bisimulations, logique temporelle, évaluation de performances
    - △ STE partitionnés produits par DISTRIBUTOR
    - △ à la volée ou séquentielle des graphes partitionnés