Approches massivement parallèle pour l'analyse de très grands espaces d'états

Christophe Joubert

EJCP'03

26 Mai – 6 Juin 2003

INRIA Rhône-Alpes

Projet VASY

INRIA RHÔNE-ALPES

INP Grenoble

Plan

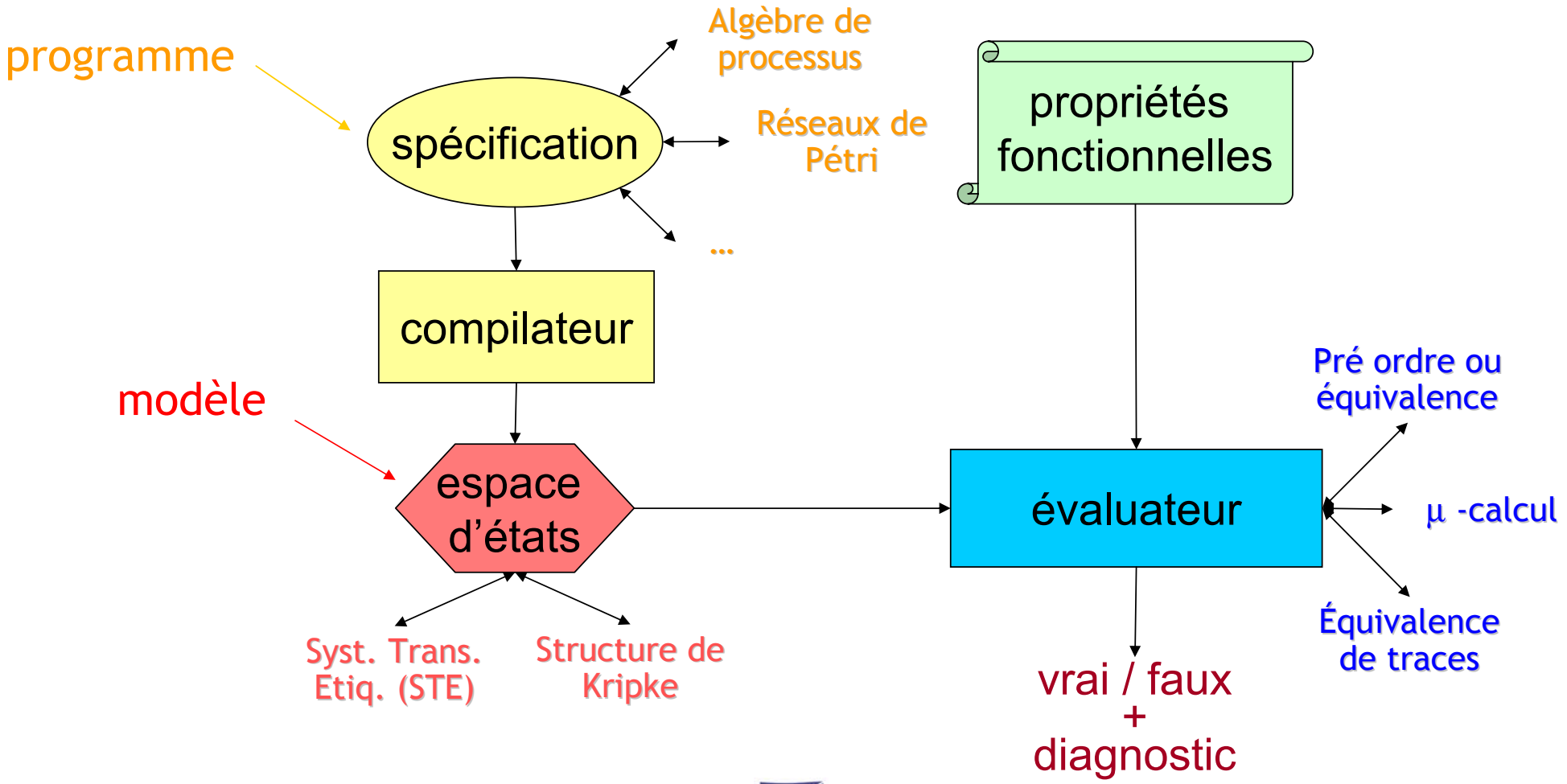
1. Quel contexte ?
2. Quels objectifs ?
3. Quelles réalisations ?
4. Quelles perspectives ?

Contexte : équipe

- Projet VASY (Validation de Systèmes) :
  - Langages et compilation
    - LOTOS, E-LOTOS, LOTOS-NT, NTIF
  - Modèles et vérification
    - Système de transitions étiquetées (STE)
    - Propriétés comportementales et logiques
- Directeur de thèse : Hubert Garavel (DR)
- Co-encadrant de thèse : Radu Mateescu (CR)

Contexte : model-checking

- Description de la vérification :



Contexte : model-checking (suite)

- Limites du model-checking
  - Explosion d'états (# états à vérifier >> capacités mémoire de la machine)
  - # travaux = améliorer les performances des algorithmes de vérification énumérative
  - Solutions proposées :
    - « à la volée »
    - Symbolique
    - Compositionnelle
    - Abstraction
    - Ordres partiels
    - Symétries
    - Réduction

Contexte : machines parallèles

- Machines parallèles prometteuses
  - Remplacer la mémoire (goulot d'étranglement) et CPU d'une seule machine par celles de plusieurs centaines (et plus) de machines
  - Utiliser à profit la présence de réseaux de stations de travail ou de grappes de PC (mis à jour régulièrement) dans tous les laboratoires, centres de recherche, entreprises
    - Grappe de 225 PC, HP e-vectra pentium III 733MHz 256 Mo
  - Etudier la faisabilité et le gain en performance de l'introduction du parallélisme dans la vérification de systèmes réels
  - Mettre à la disposition des experts en méthodes formelles des modèles, solutions, bibliothèques et outils parallèles et distribués, adaptés au problème de la vérification de systèmes
  - Tirer profit des avantages de techniques de lutte contre l'explosion d'états avec celle du parallélisme

Contexte : CADP

- CADP: une plate-forme pour l'ingénierie des protocoles et des systèmes distribués [Garavel-Lang-Mateescu-02]
  - *Construction and Analysis of Distributed Processes* (anciennement *CAESAR/ALDEBARAN Development Package)*
  - `http://www.inrialpes.fr/vasy/cadp`

- Dernière version Juillet 2001 (prochaine Juillet 2003)

- Statistiques:
  - **264 sites** sous licence (INRIA) CADP
  - Entreprises : **Bull, Ericsson, Gemplus, Lucent, Nokia, Oblog**
  - Contributions : **Verimag, PAMPA (IRISA), FMT (Uni. Twente)**
  - **58 études de cas** réalisées en utilisant CADP
  - **11 outils de recherche** construits à partir de CADP

Plan

1. Quel contexte ?
2. Quels objectifs ?
3. Quelles réalisations ?
4. Quelles perspectives ?

Model-checking réparti

- 4 Types d'analyse :
  - (questions fondamentales = place et gain escompté du parallélisme au sein de ces analyses + noyau de calcul parallèle générique commun aux quatre types d'analyses)

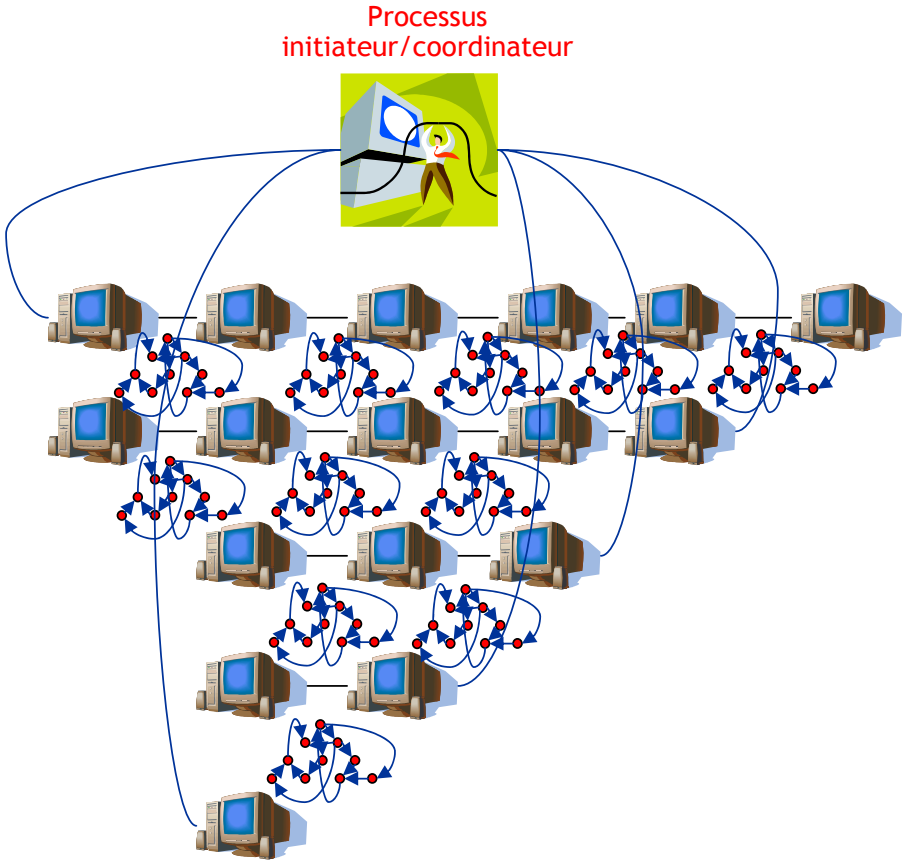
- Génération d'espace d'états [Cia01, GSM01]
  - Accessibilité (deadlock) + modèle explicite très grand
    - (goulot d'étranglement de la vérification)

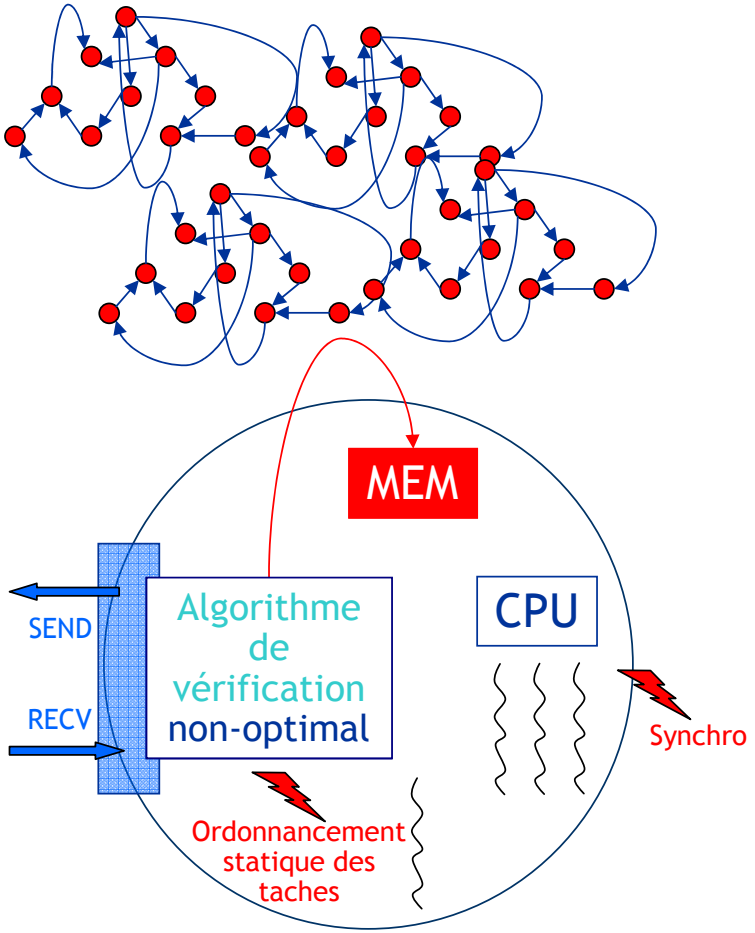
- Vérification par équivalence (bisimulation) [FM91]
  - Comparaison
  - Minimisation

- Vérification par logique temporelle
  - μ-calcul régulier d'alternance 1 (à la volée [SW89])

Model-checking réparti (suite)

- Approche suivie :





Model-checking réparti (fin)

- Approche suivie (suite):
  - Modèle implicite (Réseau de Pétri étendu et NTIF)
  - Modèle explicite (STE)
  - Parallélisme de résolution ➔ augmenter performance et taille des problèmes traités (/vs. de situation)
  - Parallélisme de contrôle ➔ modèle parallèle SPMD (famille MIMD) (/vs. de données)
  - Aspects parallèles : définition de processus, partage de tâches, équilibrage, représentation canonique, communication (+synchronisation) et terminaison
  - Types de vérification : génération de STE, fusion de STE distribués, vérification à la volée par équivalence ou par logique temporelle (μ-calcul)

Plan

1. Quel contexte ?
2. Quels objectifs ?
3. Quelles réalisations ?
4. Quelles perspectives ?

Réalisations

- Génération distribuée d'espace d'états
  - Outil DISTRIBUTOR (CADP 2003)
- Solutions apportées ou proposées :
  - Découpage algorithmique et ordonnancement des tâches
    - Priorité entre émission, calcul local, et réception
    - Thread de contrôle complètement visible
  - Bibliothèque générique de communication (et de configuration de l'environnement parallèle)
    - Passage de message par sockets non-bloquantes bufferisées sur protocoles TCP/IP
    - Gestion fine des buffers
    - Efficacité vs. Portabilité
  - Algorithmique parallèle et distribuée
    - Détection de terminaison
  - Bibliothèque générique de résolution de SEB (Système d'équations booléennes) [Mat03]
    - Algorithmes de résolution parallèle

Réalisations

- Traduction du problème de vérification vers la résolution d'un *système d'équations booléennes* (SEB) [And94]
  - Un SEB est un tuple $B = (x, M_1, \ldots, M_n)$, où
    - $x \in \boldsymbol{X}$: variable booléenne principale
    - $M_i = \{ x_j = \sigma_i \; op_j \; \boldsymbol{X}_j \}_{j \text{ in } [1, mi]}$: blocs d'équations
      - $\sigma_i \in \{ \mu, \nu \}$: signe (point fixe) du bloc i
      - $op_j \in \{ \vee, \wedge \}$: opérateur de l'équation j
      - $\boldsymbol{X}_j \subseteq \boldsymbol{X}$: variables en partie droite de l'équation j
      - F = $\vee\emptyset$ (disjonction vide) et T = $\wedge\emptyset$ (conjonction vide)
      - x_j dépend de x_k ssi $x_k \in \boldsymbol{X}_j$
      - M_i dépend de M_l ssi une x_j de M_i dépend d'une x_k de M_l
      - Bloc *fermé* : ne dépend pas d'autres blocs
  - Une relation d'équivalence vers un SEB
    - $X_{s1,s2} =_\nu (\wedge_{s1 \to a\, s1'} \vee_{s2 \to a\, s2'} X_{s1',s2'}) \wedge (\wedge_{s2 \to a\, s2'} \vee_{s1 \to a\, s1'} X_{s1',s2'})$
  - Une formule en μ-calcul régulier d'alternance 1 vers un SEB d'alternance 1 (M_i ne dépend que de $M_{i+1} \ldots M_n$)

Plan

1. Quel contexte ?
2. Quels objectifs ?
3. Quelles réalisations ?
4. Quelles perspectives ?

Perspectives

- Paralléliser la vérification :
  - par équivalence (minimisation)
  - par logique temporelle
- Prouver la correction des algorithmes distribués de vérification (accessibilité, terminaison, …) en les formalisant
- Expérimenter sur des applications réelles
  - Traiter de très larges systèmes (Giga états)
  - Démontrer l'intérêt sur des exemples industriels et des études de cas, par rapport aux autres solutions existantes pour contrer le problème d'explosion d'états
- Combiner le parallélisme avec d'autres méthodes d'analyse (compositionnelle, réduction, …)
- Optimiser les différents aspects parallèles utilisés
  - Tenir compte de la localité spatiale et temporelle (comme dans la compilation)
  - Faire du rééquilibrage dynamique de charge

Références

- [And94] H. R. Andersen. *Model Checking and Boolean Graphs*. TCS'94
- [Cia01] G. Ciardo. *Distributed and Structured Analysis Approaches to Study Large and Complex Systems*. Proc. 1st EFF/Euro Summer School on Trends in Computer Science 2001.
- [GLM02] H. Garavel, F. Lang, et R. Mateescu. *An Overview of CADP 2001*. EASST'02
- [GMS01] H. Garavel, R. Mateescu, et I. Smarandache. *Parallel State Space Construction for Model-Checking*. SPIN'01.
- [Mat03] R. Mateescu. *A Generic On-the-Fly Solver for Alternation-Free Boolean Equation Systems*. TACAS'03.
- [SW89] C. Stirling et D. Walker. *Local Model Checking in the Modal Mu-Calculus*. TASOFT'89.