



CRISS

Contrôle de
Ressources et
d'Interférence pour
Systèmes Synchrones

présenté par: Silvano Dal Zilio
responsable: Roberto Amadio

Sécurité et langages

- partage certains des problèmes liés au génie logiciel et à la vérification
- Les hypothèses changent:
 - on ne cherche pas uniquement à éliminer des erreurs, mais on considère un système sous attaque → les conditions sont plus fortes.
- Le point de vue évolue:
 - on considère des systèmes ouverts, sans infrastructure d'administration centrale

Problématique

- **Hier:** sûreté de fonctionnement; analyse à la compilation; gestion de la mémoire; ...; contrôle d'accès
- **Aujourd'hui:** code mobile (faible); vérification de la sûreté au chargement → intégrité du contexte d'exécution; inspection de la pile; *monitors*
- **Demain:** l'informatique ubiquitaire, réseaux ad-hoc, active network, ...

Problématique (demain)

- Extension à d'autres types de propriétés:
 - non interférence
 - contraintes sur l'utilisation des ressources
- Extension à d'autres langages / structures de contrôle:
 - programmation fonctionnelle
 - coordination; *threads*; *timeouts*
 - mobilité forte (i.e. pendant l'exécution)

Partenaires



1. INRIA, projet Mimosa:
non-interférence,
synchrone, mobilité
2. LIF, U. Provence:
mobilité, contrôle de
ressources, certification
logicielle (Coq)
3. LIPN, U. Villetaneuse:
complexité, typage
linéaire, inférence
4. LORIA, projet
Calligramme et Prothéo:
complexité, terminaison,
inférence

Présentation générale

- contrôle de ressources: on s'intéresse à un langage fonctionnel simple du premier ordre – et une machine virtuelle associée
 - technique qui mélange analyse de terminaison et analyse sur la taille des données
- on étudie l'extension à l'ordre supérieur
 - extension non triviale, on utilise une nouvelle approche → typage linéaire

Scénario

programmes

```
emit(e); next;  
wait s(x) in  
  f(3)  
else yield;  
stop
```

```
fun f(x: nat) =  
  g(3, x + y)  
  match z with  
  ...
```

compilation

bytecode

+
certificats

optimisation

annotations:
type, taille,
terminaison



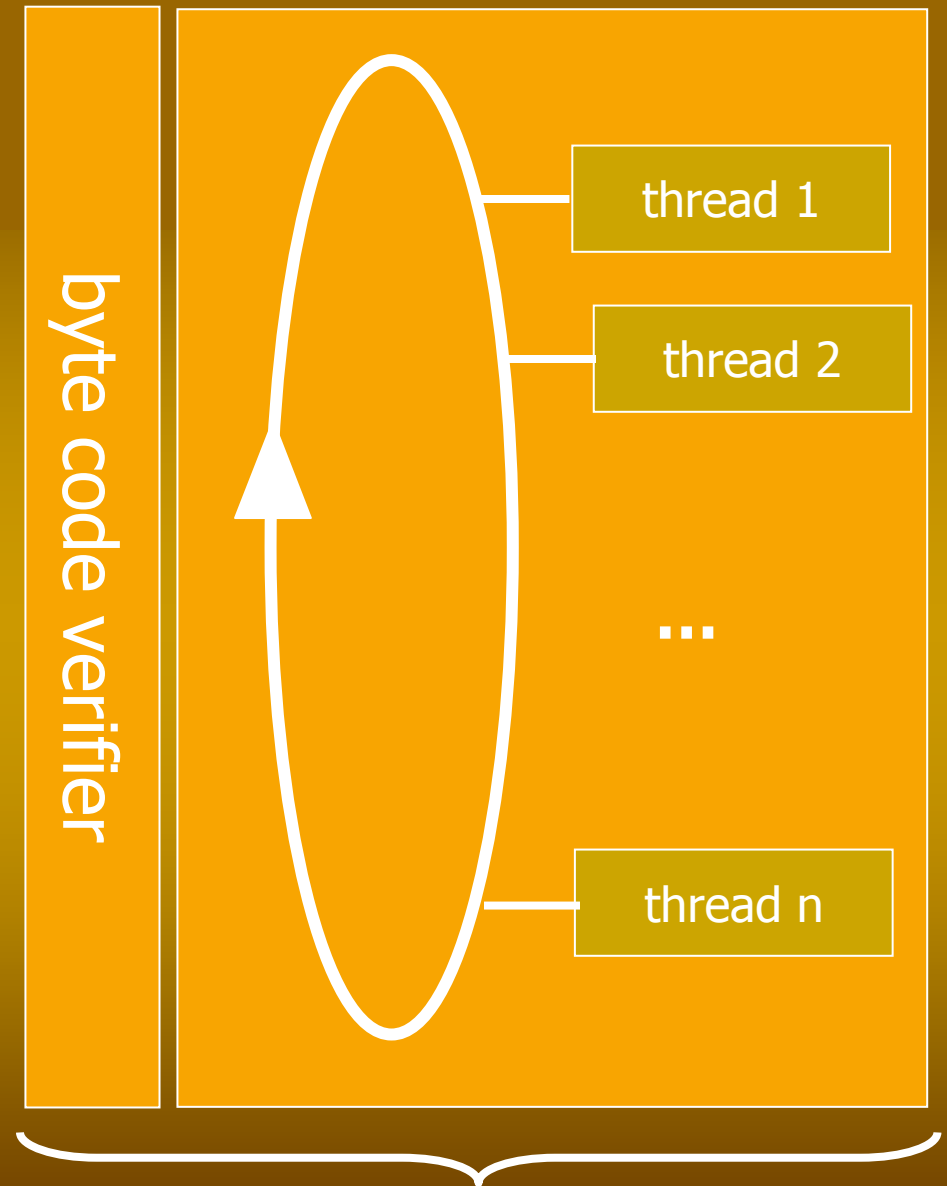
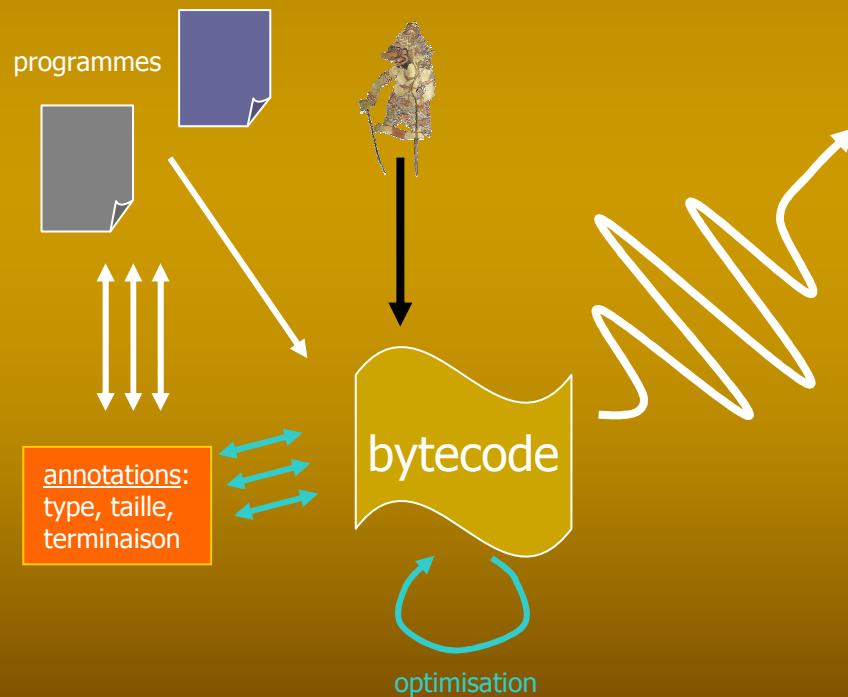
Présentation générale - suite

- on étudie les problèmes liés à la génération de certificats
 - notre approche est basée sur le typage
- on poursuit aussi un but pédagogique

Présentation générale - suite

- dans le deux cas: on s'intéresse au cas d'un langage de coordination synchrone
 - permet de tester notre approche en présence d'exécutions concurrentes: *threads coopératives*, synchronisation, évènement, *timeouts*, ...

Scénario



trusted computing base

Résultats attendus

- langage fonctionnel:
 - inférence automatique de bornes de complexité
 - ordre supérieur et typage linéaire
- langage de coordination pour systèmes synchrones:
 - conception d'un nouveau langage
 - étude des problèmes de non interférence
 - contrôle de ressources
- machines virtuelles et génération de certificats