Répartition d'applications sur Internet, sûreté et sécurité

Julien Vayssière
Julien.Vayssiere@sophia.inria.fr
http://www.inria.fr/oasis/Julien.Vayssiere

Journées PRO: Parallélisme, Répartition et Objets. Lille, 25 et 26 Novembre 1999

INRIA SOPHIA ANTIPOLIS

Activités (professionnelles) récentes

- Stage de DEA à l'INRIA Sophia-Antipolis avec Denis Caromel.
  - Conception et développement de ProActive, une implémentation 100% Java du modèle Eiffel//. Suite à cela, implication dans le Java Grande Forum.
- CSNE à Oslo en Norvège pour Schlumberger
  - Développement d'applications client-serveur en Java et CORBA
- ' Pre-Doc ' à l 'université d'Adelaide en Australie
  - Utilisation de ProActive pour implémenter des réseaux de processus de Kahn. Un peu d'enseignement aussi.
- En thèse depuis Novembre 99 à l 'INRIA Sophia-Antipolis
  - Projet OASIS, projet commun CNRS-UNSA-INRIA
  - Bourse Entreprise-Région avec la société Questel et le Conseil Régional PACA
  - Sujet : Répartition d 'applications sur Internet, sûreté et sécurité

INRIA SOPHIA ANTIPOLIS

Vers un monde mobile et connecté

- Le logiciel et le matériel sont de plus en plus mobiles
  - Code mobile: TCL, Telescript et surtout Java,…
  - Matériel mobile: ordinateurs portables, organiseurs de poche, téléphones portables, badges intelligents, cartes à puce, *Personal Area Network* d'IBM…
- Tout est connecté à tout, souvent à travers Internet
  - Le code devient mobile parce que le matériel qui l'exécute bouge

INRIA SOPHIA ANTIPOLIS

Programmer ces nouvelles applications

- L'aspect algorithmique devient secondaire
- Des problèmes connus et maîtrisés passent au premier plan:
  - Résistance aux pannes
  - Minimiser l'impact de la latence
- Des problèmes nouveaux apparaissent:
  - Réseaux fluctuants et imprévisibles (bande passante, latence, pannes)
  - Environnement ouvert et *a priori* hostile
  - Frontières administratives à traverser en permanence
- L'aspect sécurité devient un élément central de la programmation répartie

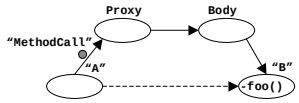
INRIA SOPHIA ANTIPOLIS

Quelle approche pour la sécurité ?

- Approche habituelle: on pense à la sécurité une fois que les fonctionnalités sont en place et qu'il est presque trop tard
  - ex: HTTP, telnet, FTP, RMI, CORBA,...
  - Solutions *ad hoc*, difficiles à évaluer et généralement sources de problèmes
- Pour des applications mobiles et connectées, la sécurité devient l'un des aspects essentiels de la conception. Pour cela il faut :
  - S'appuyer sur un modèle intégrant la sécurité depuis le départ. Ex: *Mobile Ambients* (Cardelli), *Join Calculus* (Fournier et Gonthier)
  - Avoir des fondations théoriques solides pour espérer faire des preuves
  - Une implémentation solide et relativement facile à utiliser

INRIA SOPHIA ANTIPOLIS

Lien avec ProActive

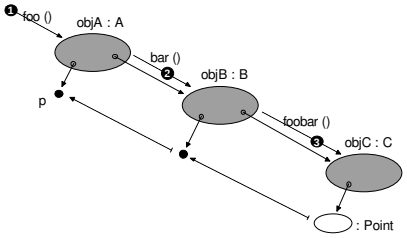


- Avec ProActive on peut
  - Implémenter la résistance aux pannes et l 'adaptation aux changements de performances du réseau dans les proxies (MOP)
  - Cacher la latence par un mécanisme de futurs et attente par nécessité
- ProActive n'intègre aucune notion de sécurité
  - Possible au niveau du couple proxy/body ?
  - Comment définir une politique de sécurité ?
  - Problème du *malicious host* avec la migration

INRIA SOPHIA ANTIPOLIS

Continuations - le problème

```
class A {
 void foo () {
  Point p = objB.bar ();
  ...
}}
class B {
  Point bar () {
    …
    return objC.foobar ();
}}
class C {
  Point foobar () {
    …
    return new Point (0,0);
}}
```

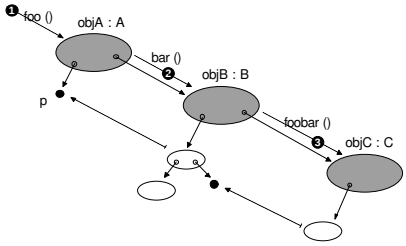


- objB délègue le calcul du résultat à objC
- Par défaut, passer un futur en paramètre ou le renvoyer comme résultat bloque si ce futur n'est pas encore disponible.
- Le blocage de objB n'est pas nécessaire
- On veut donc que **bar()** retourne avant que le futur soit disponible

INRIA SOPHIA ANTIPOLIS

Continuations automatiques

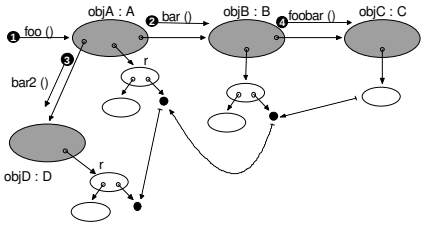
• Avec les continuations automatiques, la méthode **bar()** n'est plus bloquée en sortie quand l'objet renvoyé contient des futurs non disponibles.
• Un autre thread (non précisé) s'occupe de renvoyer le résultat à objA plus tard.
• Reste le choix de quand renvoyer le résultat à objA. Deux stratégies:
  • Renvoyer uniquement des **réponses complètes** (le graphe de l'objet renvoyé ne contient pas de futurs non encore disponibles)
  • Permettre de renvoyer des objets contenant des futurs non disponibles, il s'agit alors d 'une **réponse partielle**.



INRIA SOPHIA ANTIPOLIS

Continuations automatiques (2)

- Problème des réponses complètes: il faut parcourir l'arbre du résultat pour savoir si il n'y a plus de futurs non disponibles.
- Problème des réponses partielles: lorsque leur valeur est disponible, il faut mettre à jour les futurs en suivant une chaîne de longueur inconnue (analogie avec les problèmes de migration des objets actifs).
- Choix d'une statégie: dépend du gain amené par une réponse partielle. Ex: document composé avec certaines composantes légères (texte) et d'autres lourdes (video)
- Le problème des futurs passés en paramètres est encore plus complexe



INRIA SOPHIA ANTIPOLIS

Plan de travail

- Etude des modèles formels pour la distribution et la sécurité
- Amélioration du modèle ProActive
  - Prise en compte de la sécurité
  - Notion d'objets actifs imbriqués
  - Continuations automatiques
- Implémentation basée sur
  - Java 2: modèle de sécurité performant mais d'assez bas niveau
  - RMI: récentes améliorations pour la sécurité
  - Jini: bien adapté à des objets très mobiles

INRIA SOPHIA ANTIPOLIS