Optimiser et Répartir ses Applications Mobiles

macchiato

ADAM

http://macchiato.fr

Nicolas Petitprez, Romain Rouvoy, Laurence Duchien
ADAM, Inria Lille Nord Europe
Université Lille 1, LIFL UMR CNRS 8022

Objectifs

Créer des **applications mobiles connectées** riches en fonctionnalités en économisant les ressources du périphérique.
Macchiato permet de développer des applications mobiles **performantes**, **adaptables** et pouvant être **réparties** tout en étant économes en ressources. Cette plate-forme est agnostique vis à vis de l'environnement d'exécution. Les applications développées s'exécutent sans modification sur tous les environnements que la plate-forme supporte. Il est possible de distribuer les applications sur les navigateurs Web, sur les navigateurs Web mobiles et sur des serveurs d'applications.

Concepts

Les **acteurs** sont des entités autonomes qui communiquent par envoi de message

Les **liaisons** entre acteurs permettent aux acteurs de communiquer quel que soit leur périphérique d'exécution

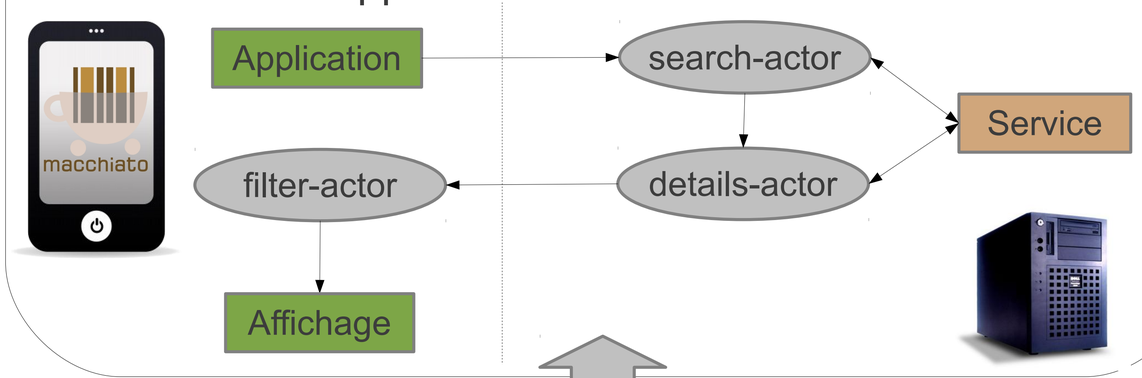
```
macchiato.deploy(function(factory) {
  factory.create("search-actor")
    .as(function(msg, context) { /*...*/ });
  factory.create("details-actor")
    .as(function(msg, context) { /*...*/ });
  factory.create("filter-actor")
    .as(function(msg, context) { /*...*/ });

  factory.bind("search-actor")
  .to("details-actor");
  factory.bind("details-actor")
  .to("filter-actor");
});
```

Macchiato propose aux développeurs un **langage de script embarqué.** Une abstraction de l'environnement permet d'avoir un code identique quel que soit l'environnement d'exécution

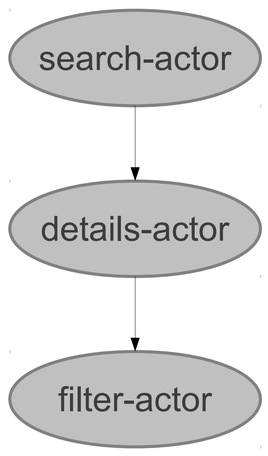
Distribution

Les acteurs sont **répartis** d'une façon **optimale** entre le smartphone et le serveur. La plate-forme est capable de redistribuer les acteurs en cas de défaillance pour garantir la **robustesse** de l'application



Introspection et Supervision

La plate-forme est **réflexive**. Elle permet **d'introspecter** l'architecture de l'application et de la **reconfigurer** par des opérations d'ajout, de modification ou de suppression d'acteurs
Elle **collecte des données** sur le fonctionnement de l'application, e.g., le nombre et la taille des messages échangés, l'utilisation de la connexion réseau, etc.



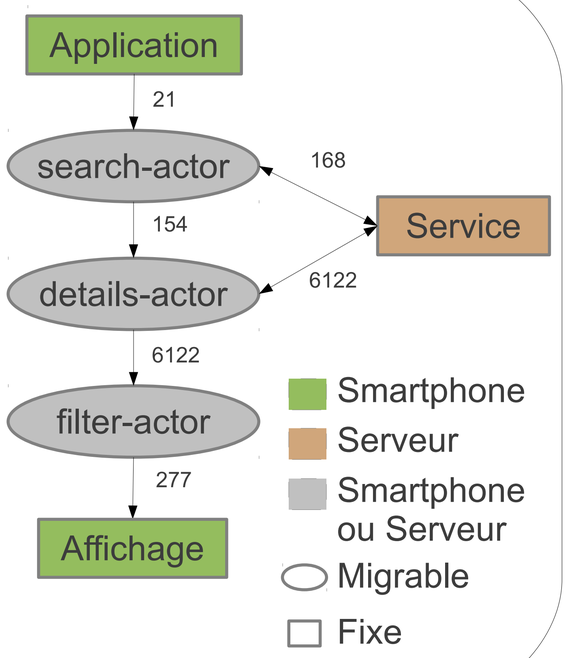
Optimisation

Les informations collectées lors de l'introspection et de la supervision permettent de construire un graphe d'exécution de l'application. Certains nœuds du graphe, correspondant à des acteurs, sont localisés : ils ne peuvent pas changer de périphérique d'exécution, d'autres sont migrables. Les arcs sont annotés avec la quantité de données échangées

Le problème d'optimisation consiste à trouver une répartition du graphe entre le smartphone et le serveur qui minimise la consommation des ressources tout en conservant le service proposé. La résolution du problème est effectuée sur le smartphone.

$$\min \left(\sum_{i=1}^{n} W_i \times \mid X_i^t - X_i^{t-1} \mid + \sum_{j=1}^{m} A_j \times \mid src_j - dest_j \mid \right)$$

- n : nombre de nœuds ;
- m : nombre d'arcs ;
- W_i : coût de migration de l'acteur i ;
- $X^t\{0,1\}$: position calculée du nœud, 0 : smartphone, 1 : serveur ;
- $X^{t-1}\{0,1\}$: position précédente du nœud, 0 : smartphone, 1 : serveur ;
- A_j : coût d'utilisation de l'arc j ;
- $src_j\{0,1\}$: position calculée du nœud au début de l'arc j ;
- $dest_j\{0,1\}$: position calculée du nœud à la fin de l'arc j ;



Inria informatiques mathématiques
cnrs dépasser les frontières
Université Lille1 Sciences et Technologies
lifl Laboratoire d'Informatique Fondamentale de Lille