

# Make Distribué dRuby / Rinda

ASSOULINE Daniel

GIROUX Baptiste

SIBILLE Gaspard

TAVERNIER Vincent

# Technologies utilisées

## Code de l'application

### **Rinda**

Implémentation des « tuple spaces »

### **dRuby**

Système d'objets distribués

### **Ruby**

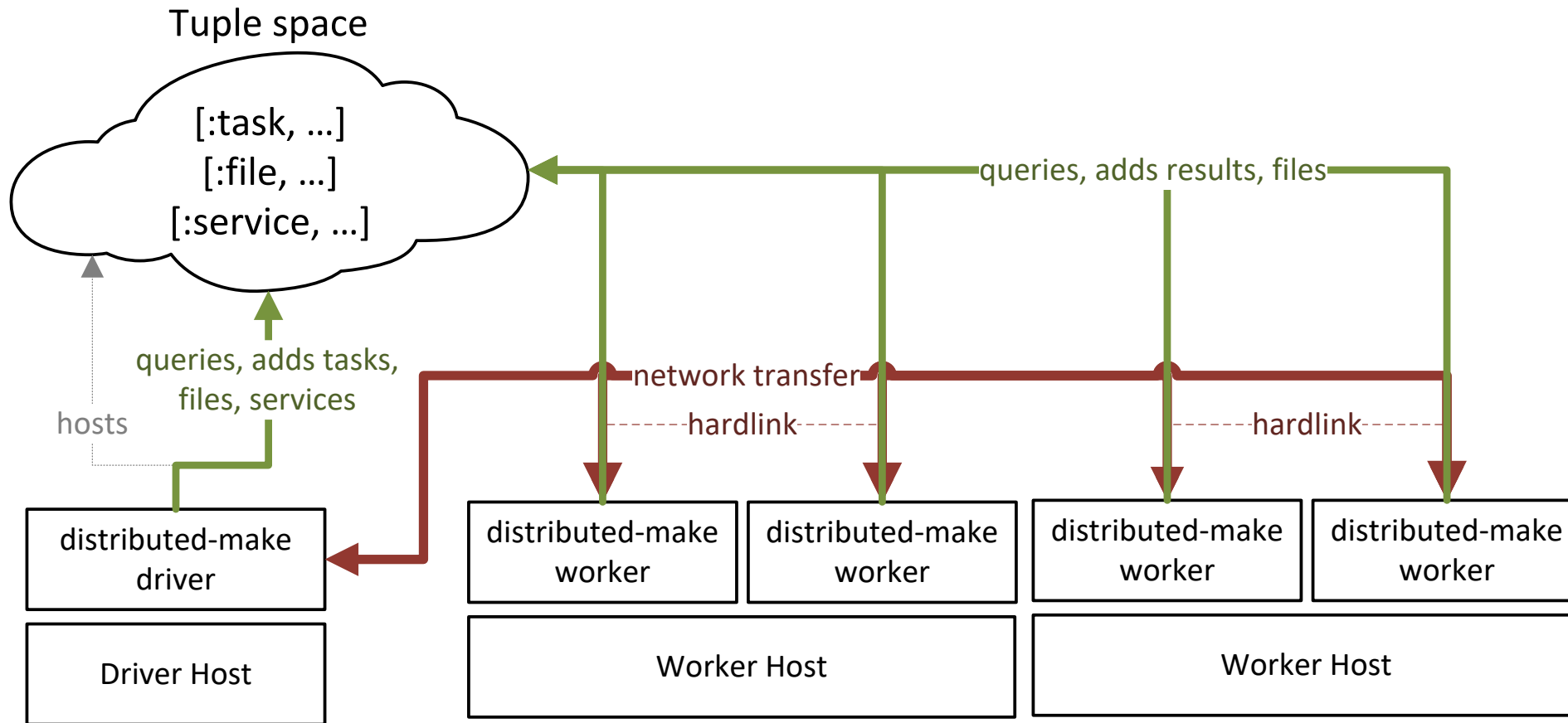
Langage de script dynamique orienté objet

Make Distribué - dRuby/Rinda

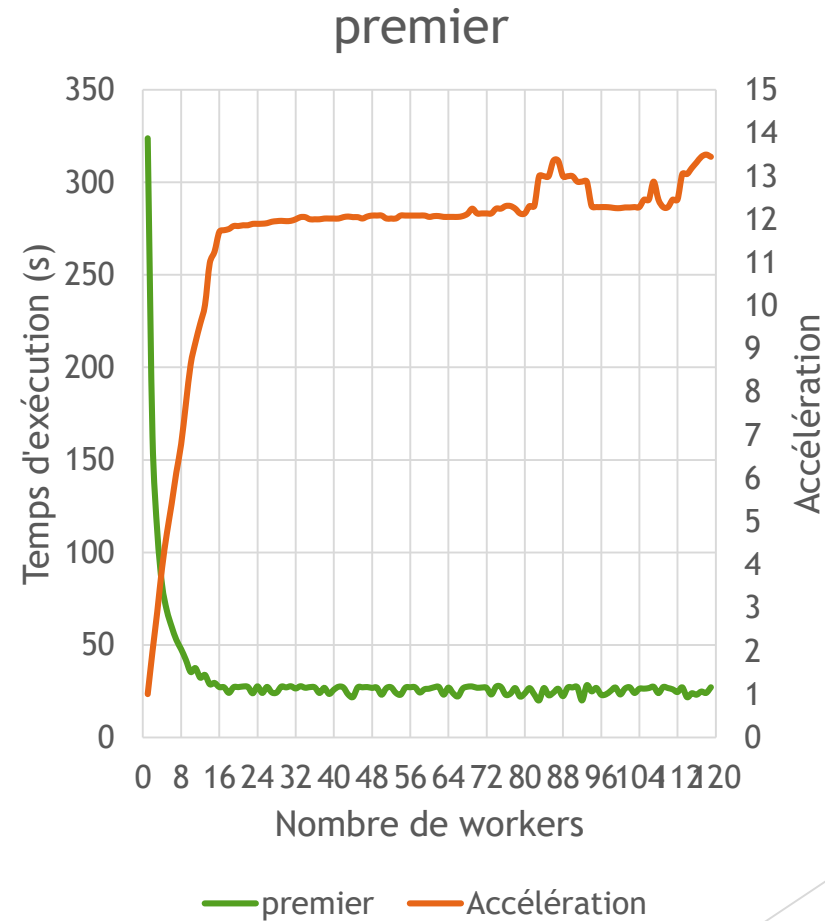
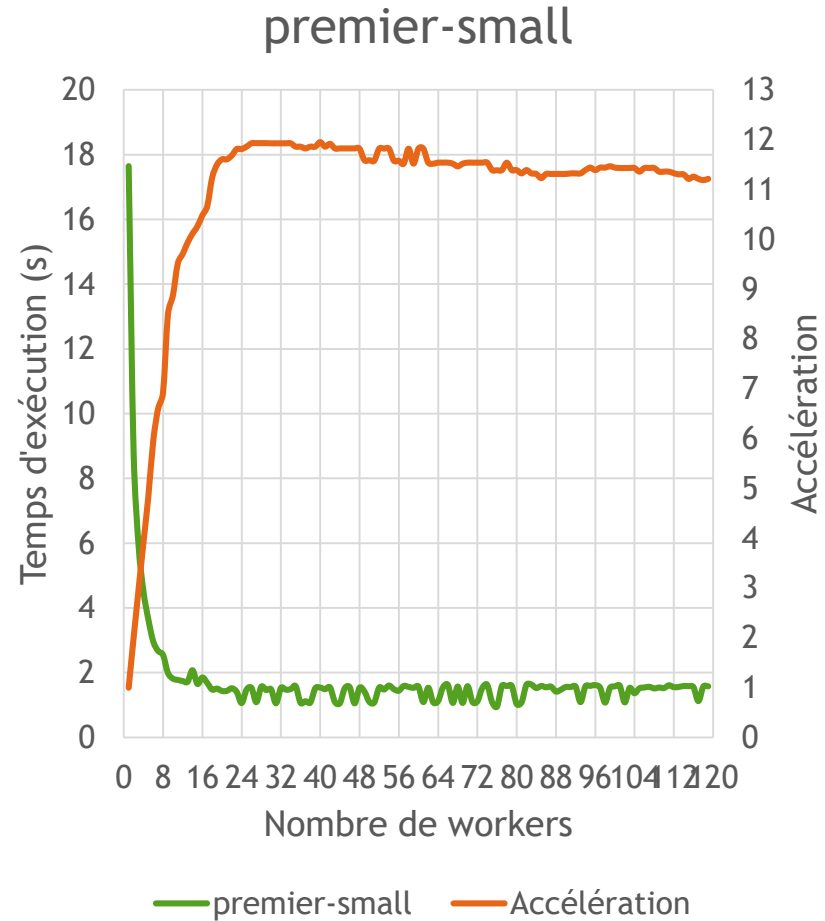
## Test et déploiement

- ▶ Environnement de test : Vagrant
- ▶ Déploiement simultané : SSHKit
- ▶ Documentation du code : YARD
- ▶ Tests unitaires : Rspec
- ▶ Profileur : ruby-prof
- ▶ Task runner : Rake

# Architecture du système

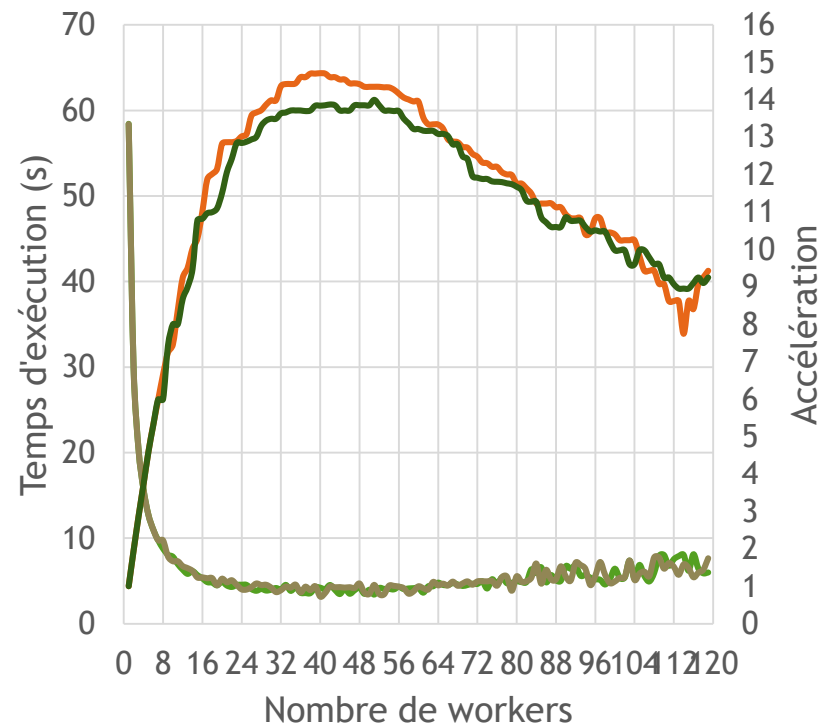


# Résultats (1)



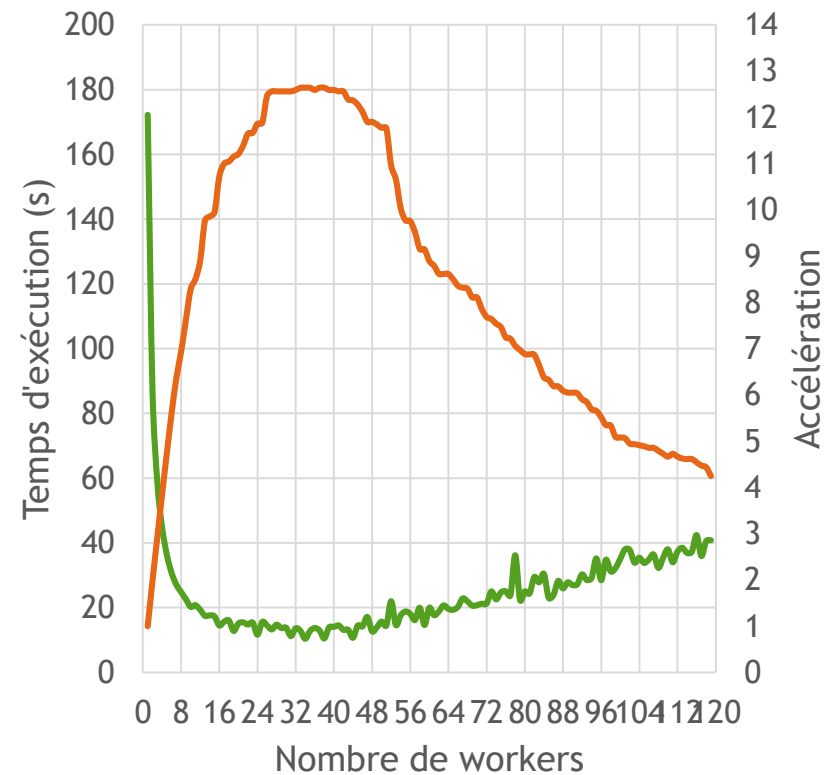
# Résultats (2)

blender-2.49



blender-2.49  
blender-2.49-recuse  
Accélération  
Accélération (recuse)

blender-2.59



blender-2.59  
Accélération

# Conclusions

- ▶ Rinda : abstraction d'un espace partagé entre processus assurant de nombreuses propriétés (si bien utilisé)
  - ▶ dRuby : simple à implémenter, manipulation transparente des objets distants
  - ▶ Ruby : grande souplesse due à l'aspect dynamique du langage, beaucoup d'outils et de librairies disponibles
- 
- ▶ Rinda : risque de deadlock si mal utilisé ; un seul processus gère tout l'espace
  - ▶ dRuby : bibliothèque aux performances discutables (l'établissement de connexions TCP introduit de la latence)
  - ▶ Ruby : parallélisme peu performant à cause du GIL (Global Interpreter Lock)