

# Un service distribué sur le cloud pour l'exécution de chaînes de traitements sur la grille

Tristan Glatard

Université de Lyon, CREATIS; CNRS UMR5220; Inserm U1044; INSA-Lyon; Univ. Lyon 1, France

## Introduction

### Exécution distribuée de chaînes de traitements (*workflows*)

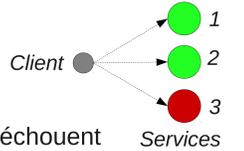
- Eviter point unique de défaillance
- Garantir le passage à l'échelle

### Distribution par *round-robin*

- Sensible aux pannes
- Pas de contrôle de charge

### Exemple

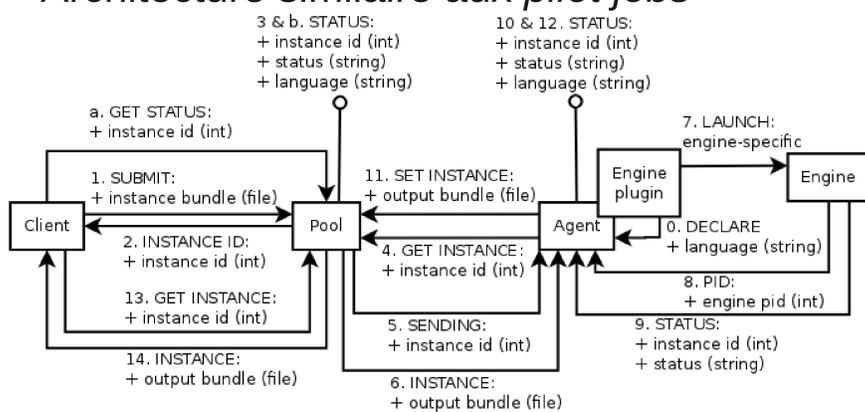
Un tiers des soumissions échouent  
Les services 1 et 2 reçoivent la même charge



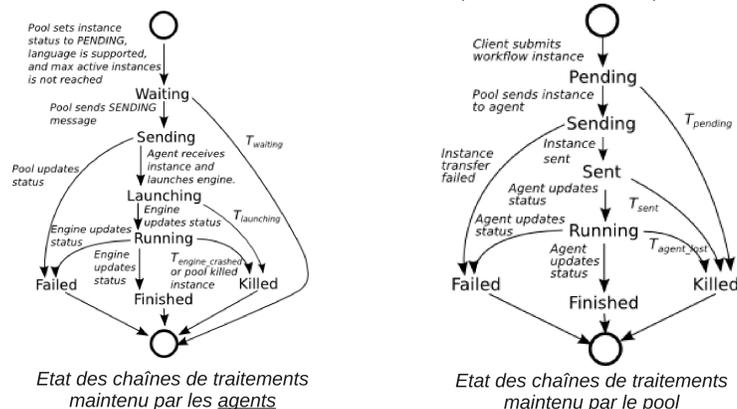
## Méthode et implémentation

### Architecture proposée

- *Pool* central
- Agents *workflow executors* distribués
- Agents se connectent en fonction de leur état et charge
- Architecture similaire aux *pilot jobs*



Architecture du service distribué d'exécution de chaînes de traitements. La numérotation décrit les étapes de l'exécution d'une chaîne de traitements. Les flèches terminées par des cercles correspondent à des broadcast.



Etat des chaînes de traitements maintenu par les agents

Etat des chaînes de traitements maintenu par le pool

### Implémentation

- Messages échangés par XMPP
- Plugins utilisent JSPF

## Conclusion

- Distribution de chaînes de traitements par approche similaire aux *pilot jobs*
- Assure robustess et passage à l'échelle
- Les agents contrôlent leur charge
- *Cloud* adapté au déploiement d'agents
- Perspective : déploiement en production, *Virtual Imaging Platform* - <http://vip.creatis.insa-lyon.fr>

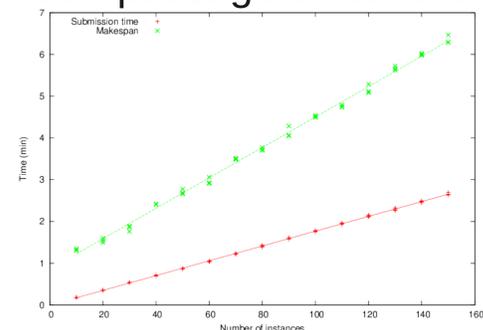
## Expériences et résultats

### Déploiement

- Agents et moteurs d'exécution sur StratusLab (LAL)
- Chaînes de traitements exécutées avec MOTEUR
- *Pool* et clients sur le même réseau
- $T_{sent} = 30s$ ,  $T_{agent\_lost} = 10s$ ,  $T_{engine\_crashed} = 3s$ ,  $T_{launching} = 5s$

### Exp1 : passage à l'échelle

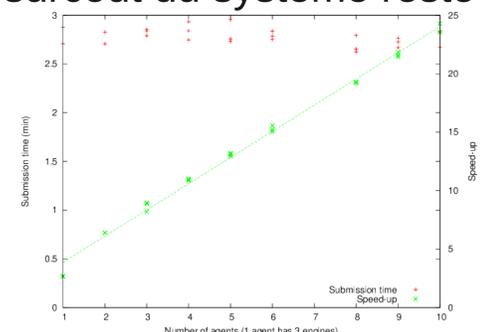
- Exp 1-a : nombre de chaînes de traitements
- 10 agents, 3 exécutions concurrentes par agent
- De 10 à 150 chaînes de traitements exécutées
- Résultats : bon passage à l'échelle



Exp1-a : passage à l'échelle en fonction du nombre d'exécutions concurrentes

### Exp1-b : nombre d'agents

- 150 chaînes de traitements exécutées
- De 1 à 10 agents
- Résultat : le surcoût du système reste contrôlé



Exp1-b : passage à l'échelle en fonction du nombre d'agents

### Exp2 : robustesse

- Deux agents déployés
- Flapping : déconnexion d'un agent (pendant 5s toutes les 10s)
- Crash : crash d'un agent après 90s
- Résultat : totalement robuste à Flapping

### impact limité de Crash

Test	Test		Flapping		Crash			
	#Killed	Makespan(s)	#Killed	Makespan(s)	#Killed	Makespan(s)		
#1	0	321	#1	0	319	#1	3	454
#2	0	318	#2	0	326	#2	3	385
#3	0	317	#3	0	319	#3	3	454

Exp 2 : robustesse à la perte temporaire ou totale de connexion avec les agents