

Gru: an Approach to Introduce Decentralized Autonomic Behavior in Microservices Architectures

Luca Florio, Elisabetta Di Nitto - 2016

Doryan Bonifassi, Florian Bourniquel, François Melkonian



Contexte

- De plus en plus de systèmes basés sur des micro-services
- Ces systèmes sont de plus en plus conteneurisés
- Comment faire pour que ces conteneurs s'auto-gèrent ?



Objectifs du papier de recherche

- Créer une sur-couche à docker, des agents GRU
- Automatiser les décisions sur la *scalabilité*
- *Framework* pour définir le comportement des agents
- Distribution du service registry pour éviter le *SPOF*
- Déploiement de plusieurs services sur machine
- Mise en place d'indicateurs orientés métier



Fonctionnement

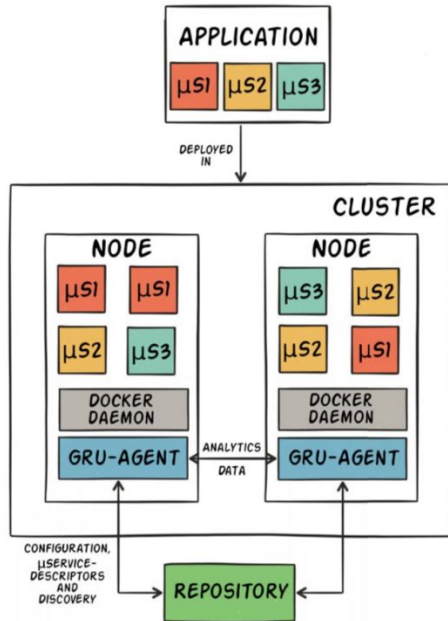


Fig. 1. Gru deployment with two nodes.

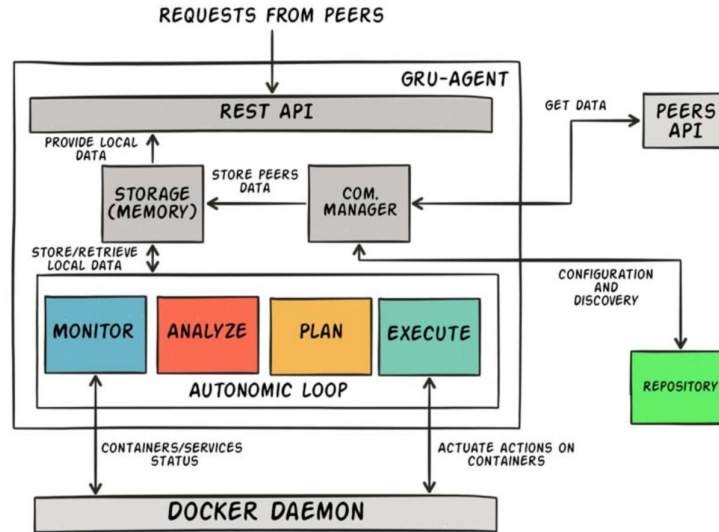


Fig. 2. Gru-Agent components and their interaction.



Fig. 3. Requests to the service (mean 1 minute).



Fig. 5. Active instances of service during time.



Validité

- Seulement exécuté 5 fois sur un “petit” projet
- Seulement 2 machines



Analyse - Inconvénients

- Duplication des dépendances entre modules
- Coût du système global
- L'algorithme aléatoire peut entraîner des coûts



Analyse - Avantages

- Réaction rapide face à un pique de charge
- Élasticité verticale et horizontale automatique sur docker
- Bonnes capacités d'évolution



Prise de recul

