

Allocation de ressources par une méthode hybride machine learning - optimisation en contexte de conteneurs

Etienne Leclercq ^{*,**} Jonathan Rivalan ^{*}, Frédéric Roupin ^{**}, Céline Rouveirol ^{**}

^{*}Alter way - (etienne.leclercq,jonathan.rivalan)@alterway.fr,

^{**}LIPN - (leclercq, roupin, rouveirol)@lipn.univ-paris13.fr

1 Introduction

La gestion des ressources est l'un des défis les plus importants du cloud et est composé principalement du problème de placement des services sur l'infrastructure, défi difficile notamment par l'introduction de l'élasticité. Ceci rend la modélisation du problème d'affectation des conteneurs complexe, car cette élasticité entraîne de l'inconnu avant l'affectation. Lorsque l'on cherche une bonne solution pour le placement des conteneurs, l'objectif global est de limiter la surcharge des serveurs qui se traduit par des interruptions de services, et d'optimiser l'utilisation globale des ressources, comme le nombre de serveurs en cours d'utilisation. Pour atteindre ces objectifs, nous considérons 4 critères qui représentent le nombre de nœuds, l'amplitude et la variance de la consommation des nœuds, et le nombre de dépassement de capacité des nœuds. En pratique, le scheduler de Docker ne tient pas compte de l'utilisation des ressources pour placer les conteneurs dans un cluster (Hassen, 2016). D'autres travaux se sont intéressés au placement de conteneurs (Tosatto et al., 2015), mais sans possibilité de l'ajuster en fonction de l'évolution des consommations. L'ajustement des ressources allouées aux conteneurs a également été étudié, mais sans modifier leur placement sur l'infrastructure. Pour répondre à cette problématique et les objectifs cités, nous avons développé une méthode alliant apprentissage et optimisation. Cette interaction constitue par ailleurs un intérêt supplémentaire, étant encore peu développée aujourd'hui, un domaine étant souvent utilisé pour améliorer les méthodes de l'autre. Ici, le clustering est utilisé pour identifier les profils de consommation et guider le placement en utilisant des techniques d'optimisation.

2 Méthodologie - la boucle de rétro-action

Comme en production, de nouvelles données sont collectées en continu, nous avons développé un mécanisme de *boucles de rétro-action*, qui vise à évaluer la solution actuelle avec de nouvelles données, et à mettre à jour cette solution si besoin. Par ailleurs, les services fonctionnant en temps réel, la solution doit être trouvée rapidement, la modélisation du problème complet avec tous ses critères et contraintes ne peut être envisagée. Ainsi nous utilisons des informations de clustering, mises à jour dans le temps grâce à des méthodes d'optimisation, pour évaluer la solution actuelle d'affectation et la mettre à jour si besoin. Les problèmes de clustering et de placement sont donc modélisés par des problèmes d'optimisation, dont les relaxations continues sont considérées. A ces problèmes nous ajoutons des contraintes de type *mustLink* pour indiquer les solutions actuelles, où $u_{i,j}$ et $v_{i,j}$ sont des variables spécifiant si i

et j appartiennent au même cluster (resp. noeud) dans le clustering C_t (resp. placement A_t) :

$$\begin{cases} \text{mustLink}_C : u_{i,j} = 1 \forall (i,j) \text{ appartenant au même cluster dans } C_t \\ \text{mustLink}_A : v_{i,j} = 1 \forall (i,j) \text{ appartenant au même noeud dans } A_t \end{cases} \quad (1)$$

Nous utilisons les variables duales associées à ces contraintes (grâce à la résolution quasi-immédiate des relaxations continues) pour obtenir les informations des modifications dans un premier temps du clustering (en fonction de l'évolution des profils de consommation) puis du placement (en fonction de l'évolution du clustering). La solution du clustering est également incorporée dans l'objectif du placement pour guider la solution de ce dernier grâce aux profils. Les variations des valeurs des variables duales d'une itération par rapport à la précédente nous indique les possibles conflits. Nous ré-assignons les individus en conflit respectivement au cluster le plus proche (via le profil moyen) et au noeud résultant de la variance totale de consommation la plus faible. Comme notre boucle fonctionne en tout temps avec des solutions déjà existantes, les solutions initiales sont fournies grâce à des heuristiques, d'une part de clustering (e.g. algorithme K-means) d'autre part via une heuristique de placement utilisant les clusters créés à l'étape précédente, en co-localisant les clusters les plus distants.

3 Evaluation

Pour mesurer l'efficacité de notre méthode, ainsi que la modification itérative du clustering pour le placement, nous avons réalisé des tests sur 2 jeux de données réels différents, qui proviennent d'Alibaba (Alibaba, 2018) et d'Alter way, partenaire industriel du projet. Ces données représentent respectivement 7 et 8 jours de consommation pour plus de 300 conteneurs. Nous avons comparé notre méthode à plusieurs méthodes, dont le spread utilisé par défaut par Docker ou le placement initial des jeux de données. Le tableau 1 montre les résultats moyens, sur des tests faisant varier le nombre de clusters et la durée des périodes d'observation, pour les 4 critères relevés, et montre pour notre méthode une économie de serveurs tout en gardant des indices de stabilité proches des méthodes utilisant plus de noeuds.

Method	c_1 nodes	c_2 ampli.	c_3 variance	c_4 overloads	Method	c_1 nodes	c_2 ampli.	c_3 variance	c_4 overloads
Initial	18	7,52	1,13	0	Initial	3	3,90	1,73	0
Spread	18	7,58	1,14	0	Spread	3	3,82	1,72	0
IC	16,25	11,5	1,35	4,5	IC	2	4,10	2,15	2,85
Heuristic	16	9,96	1,32	3,25	Heuristic	2	4,74	2,19	1,69
Loop	16	8,08	1,31	0,63	Loop	2	4,09	2,08	0,15
Loop_kmeans	16	8,92	1,34	1,17	Loop_kmeans	2	4,53	2,16	0,77

TAB. 1 – Valeurs moyennes pour les 4 critères pour Alibaba (gauche) et Alter way (droite).

Références

- Alibaba (2018). Alibaba cluster trace program. Available at <https://github.com/alibaba/clusterdata>.
- Hassen, A. (2016). A survey of docker swarm scheduling strategies.
- Tosatto, A., P. Ruiu, et A. Attanasio (2015). Container-based orchestration in cloud : State of the art and challenges. In *2015 Ninth International Conference on Complex, Intelligent, and Software Intensive Systems*, pp. 70–75.