Stage M2 2026

Ordonnancement de tâches de streaming sur des instances Cloud burstables

Superviseurs : Daniel Wladdimiro, Alessio Pagliari
Contacts : daniel.wladdimiro@davinci.fr, alessio.pagliari@lip6.fr
Lieu : Ecole Supérieure d'Ingénieurs Léonard de Vinci (ESILV), 12 Av. Léonard de Vinci, 92400 Courbevoie

Début : Environ février/mars 2026
Durée : 6 mois

Contexte

Les plateformes de cloud (comme AWS, GCP, Azure) proposent différents types d'instances de calcul, chacune avec un compromis entre performance et coût. Les instances standards offrent une performance CPU stable, mais peuvent être coûteuses pour des charges de travail variables. Une alternative économique est l'utilisation d'**instances burstables** (p.ex., les séries T d'AWS[1].

Ces instances accumulent des « crédits » CPU lorsqu'elles sont peu utilisées et les dépensent pour fournir des pics (des « bursts ») de performance lorsque la charge augmente. Ce modèle est idéal pour des applications dont l'activité est sporadique. Cependant, une fois les crédits épuisés, la performance CPU est fortement limitée, ce qui peut dégrader la qualité de service pour des tâches exigeantes.

Les **systèmes de traitement de flux** (Stream Processing Systems, SPS), comme Flink ou Spark Streaming, traitent des données en temps réel. Leurs charges de travail sont souvent dynamiques et imprévisibles, ce qui en fait des candidats intéressants pour une exécution sur des instances burstables afin de réduire les coûts [1].

Motivation et défis

L'objectif principal est de réduire les coûts d'infrastructure pour le traitement de flux en exploitant intelligemment les instances burstables [2], mais leur nature contrainte introduit plusieurs défis : **Risque de dégradation de performance** — éviter que des tâches critiques (sensibles à la latence) ne s'exécutent sur des instances ayant épuisé leurs crédits CPU ; **Hétérogénéité des tâches** — les opérateurs ont des profils variés (certains périodiques et légers, d'autres continus et intensifs) ; **Ordonnancement et placement** — un ordonnanceur classique ignore le mécanisme de crédits et peut placer une tâche CPU-intensive sur une instance proche de la limitation, augmentant la latence. Le défi est donc de concevoir un **ordonnanceur conscient des spécificités des instances burstables** pour allouer les tâches de façon optimale.

Objectifs

Le but du stage est de concevoir, prototyper et évaluer un ordonnanceur pour des tâches de streaming sur un cluster d'instances burstables.

Le stage pourra suivre ces étapes :

- **État de l'art :** Étudier le fonctionnement des instances burstables chez les principaux fournisseurs cloud (AWS, GCP, Azure) et analyser les stratégies d'ordonnancement existantes pour les systèmes de traitement de flux [3].
- **Mise en place d'une plateforme :** Déployer un SPS open-source (p.ex., Apache Flink) sur un cluster d'instances burstables (via des crédits académiques ou un simulateur).
- **Conception et prototypage de l'ordonnanceur :** Développer une politique d'ordonnancement qui prend en compte l'état des crédits CPU et les caractéristiques des tâches.
- **Expérimentations :** Évaluer l'approche sur une application de streaming représentative. Les métriques clés seront la **latence de traitement**, le **débit** et, bien sûr, le **coût** de l'infrastructure. L'évaluation comparera notre ordonnanceur à la stratégie par défaut du SPS.

Prérequis

Le candidat doit maîtriser au moins un langage de programmation (Java, Scala ou Python), disposer de bases en systèmes d'exploitation (ordonnancement, processus) et en cloud computing, ainsi que de notions en systèmes distribués. Idéalement, il a déjà été exposé à un fournisseur cloud (AWS, GCP, Azure) et connaît les principes des systèmes de traitement de flux (Flink, Spark Streaming, etc.), compétences qui pourront être approfondies durant le stage.

[1] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/burstable-performance-instances.html

Bibliography

[1] D. Wladdimiro, A. Pagliari, and R. C. Brum, "Toward Stream Processing Efficiency Leveraging Cloud Burstable Instances," in *2025 IEEE International Conference on Cloud Engineering (IC2E)*, 2025.

[2] R. Pinciroli, A. Ali, F. Yan, and E. Smirni, "CEDULE+: Resource management for burstable cloud instances using predictive analytics," *IEEE Transactions on Network and Service Management*, 2020.

[3] M. Nardelli, V. Cardellini, V. Grassi, and F. L. Presti, "Efficient operator placement for distributed data stream processing applications," *IEEE Transactions on Parallel and Distributed Systems*, 2019.