Négociation pour la consommation adaptative d'allocation continue

Ellie Beauprez[1], Anne-Cécile Caron[1], **Maxime Morge**[2], Jean-Christophe Routier[1]

[1]Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL
[2]Univ. Lyon, UCBL, CNRS, INSA Lyon, UMR 5205 LIRIS

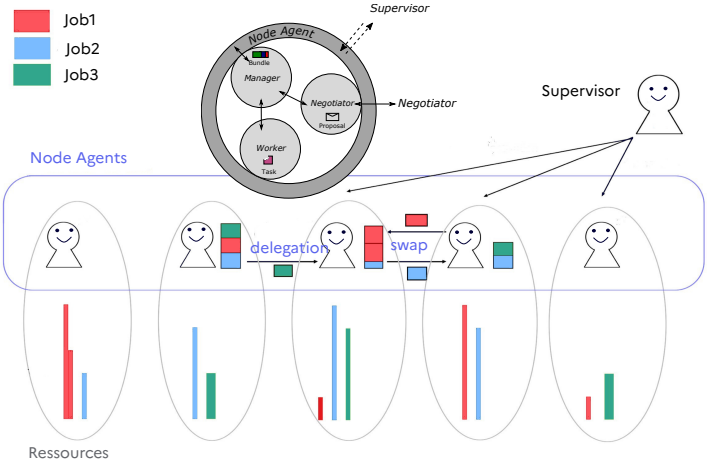
26 mars 2025

Consommation adaptative par négociation continue

- **Application pratique.**
  Déploiement du modèle de conception MapReduce pour le traitement de données massives sur un cluster
- **Objectif.**
  Répartir la charge du traitement des ressources situées sur les nœuds afin d'exécuter les tâches concurrentes le plus rapidement possible
- **Proposition.**
  Une architecture modulaire d'agent permettant la simultanéité entre négociation et consommation
- **Contributions.**
  1. Formalisation du critère de rationalité des échanges locaux qui garantit la terminaison du processus
  2. Architecture modulaire d'agent composée d'un exécutant, d'un négociateur et d'un gestionnaire
  3. Expérimentations approfondies qui montrent l'adaptation à la libération de jobs et aux aléas d'exécutions malgré une connaissance imparfaite de l'environnement

Vue d'ensemble : équilibrage de la charge

Chaque agent-nœud représente un nœud de calcul en gérant son lot de tâches



Plan

1. Travaux connexes
2. Formalisation
   - Problème STAP
   - Consommation et réallocation
   - Critère de rationalité
3. Méthode multi-agents
   - Modèle des pairs
   - Phases de négociation
4. Architecture d'agent
   - Architecture composite
   - Comportements
5. Expérimentations
   - Réallocation instantanée
   - Réallocation continue
6. Conclusion

Travaux connexes

Une approche centrée individus avec des connaissances imparfaites et sans phase d'apprentissage

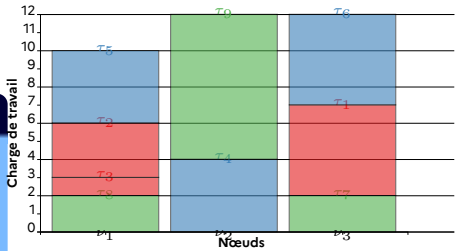
- Caractéristiques du problème :
  - tâches mono-exécutant, indépendantes, non divisibles et non préemptables
  - exécutants multi-tâches
  - ressources partageables et duplicables
- Limites des solutions centralisées [Jiang, 2016] :
  1. impossibilité de traiter des problèmes à grande échelle
  2. faible réactivité aux changements
- Beaucoup de travaux sont basés sur des enchères [Choi et al., 2009; Koenig et al., 2006] mais :
  - nous considérons des **agents coopératifs** visant à minimiser le *flowtime* moyen, i.e. la durée moyenne entre la date de libération des jobs et leur date de réalisation
  - nous préférons un **protocole bilatéral** qui permet plusieurs négociations concurrentes
- Dans la lignée de Baert [2019]
- Contrairement à [Chen et al., 2016], les agents peuvent avoir une connaissance imparfaite de l'environnement
- Contrairement à [Creech et al., 2021], ma solution ne requiert aucun modèle préalable sur les données ou l'environnement, ni phase d'exploration

Formalisation

Définition : STAP

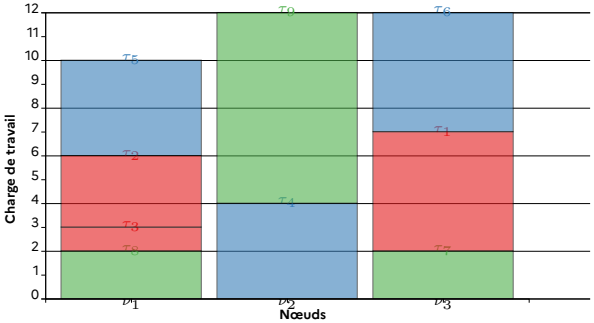
Un **problème d'allocation de tâches situées** (STAP - *Situated Task Allocation Problem*) à l'instant t est un quadruplet STAP $= \langle \mathcal{D}, \mathcal{T}_t, \mathcal{J}_t, c \rangle$ où :

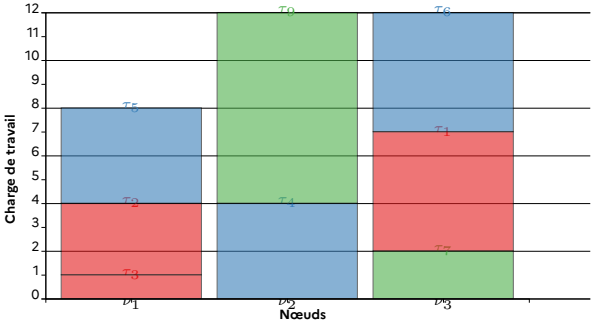
- $\mathcal{D}$ est un système distribué avec m nœuds
- $\mathcal{T}_t = \{\tau_1, \ldots, \tau_n\}$ est un ensemble de n tâches
- $\mathcal{J}_t = \{J_1, \ldots, J_\ell\}$ est un partitionnement des tâches en ℓ jobs
- $c : \mathcal{T}_t \times \mathcal{N} \mapsto \mathbb{R}_+^*$ est la fonction de coût



- ***Flowtime* moyen** = durée moyenne de réalisation des jobs pour une allocation
- $C_{J_{rouge}}(\overrightarrow{A_t}) = 7$
- $C_{J_{bleu}}(\overrightarrow{A_t}) = 12$
- $C_{J_{vert}}(\overrightarrow{A_t}) = 12$
- Soit $C_{mean}(\overrightarrow{A_t}) \simeq 10,33$

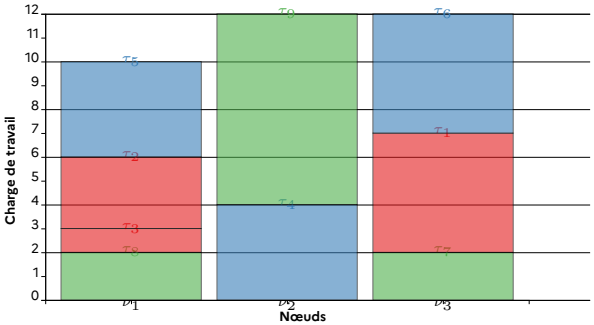
Opérations : consommation

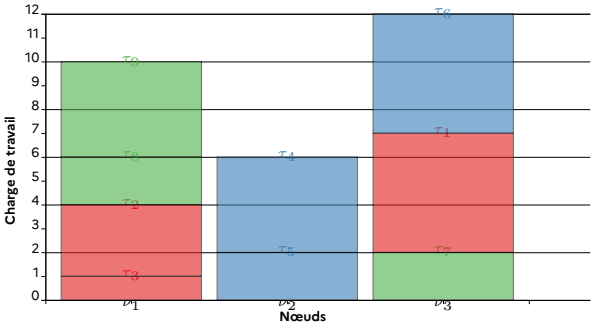




Le nœud ν_1 consomme la tâche τ_8

Opérations : réallocation bilatérale





Les nœuds ν_1 et ν_2 échangent les tâches τ_5 et τ_9
Flowtime avant : $10,33$
Flowtime après : $9,66$

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8	τ_9
ν_1		✓	✓					✓	✓
ν_2				✓	✓				
ν_3	✓					✓	✓		

Critère de rationalité

Définition : réallocation socialement rationnelle

Soit $\overrightarrow{A_t}$ une allocation pour un problème STAP. La réallocation bilatérale $\gamma(T_1, T_2, \nu_i, \nu_j, \overrightarrow{A_t})$ est **socialement rationnelle** si et seulement si *flowtime* diminue strictement,

$$\underbrace{C_{mean}(\gamma(T_1, T_2, \nu_i, \nu_j, \overrightarrow{A_t}))}_{\text{flowtime après réallocation}} < \underbrace{C_{mean}(\overrightarrow{A_t})}_{\text{flowtime de l'allocation courante}} \quad (1)$$

Propriété : terminaison

Toute séquence de réallocations socialement rationnelles est finie.

Modèle multi-agents

Prise de décision locale

- Un agent a :
  - des connaissances précises sur ses jobs et ses tâches
  - des croyances sur les jobs de ses pairs construit à partir des messages échangés
  - aucune information sur l'allocation des tâches chez ses pairs
- Les agents présument qu'ils ont tous la même stratégie de consommation

Définition : acceptabilité

Soit $\overrightarrow{A_t}$ une allocation pour un problème STAP. La réallocation bilatérale $\gamma(T_1, T_2, \nu_i, \nu_j, \overrightarrow{A_t})$ est **acceptable** par l'agent ν_i ssi il croit que le *flowtime* diminue strictement,

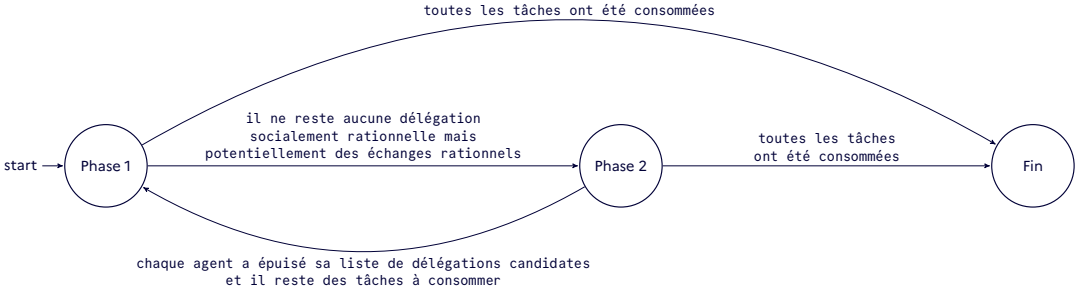
$$\Sigma_{J \in \mathcal{J}} \max_{\forall \nu_o \in \mathcal{N} \setminus \{\nu_i, \nu_j\}} (\underbrace{C_J(\overrightarrow{B_{i,t} \ominus T_1 \oplus T_2})}_{\text{connaissance locale}}, \underbrace{C_J{}^i(\overrightarrow{B_{j,t} \ominus T_2 \oplus T_1})}_{\text{croyance sur le contractant}}, \underbrace{C_J{}^i(\overrightarrow{B_{o,t}})}_{\text{croyance sur les autres pairs}}) < C^i_{mean}(\overrightarrow{A_t}) \quad (2)$$

Négociation en deux phases

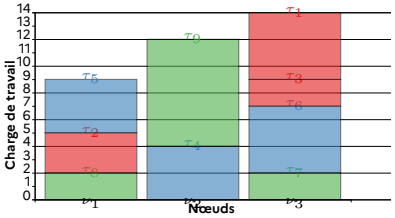
Comme chercher une contre-offre peut être coûteux, nous utilisons des échanges pour sortir des minima locaux

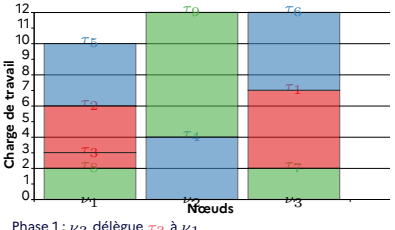
- **Phase 1.** Le donneur propose des délégations socialement rationnelles
- **Phase 2.** Le donneur propose des délégations même si elles ne sont pas rationnelles, et le receveur cherche une contre-offre pertinente

Alternance entre les deux phases jusqu'à ce qu'il n'y ait plus de tâche à consommer

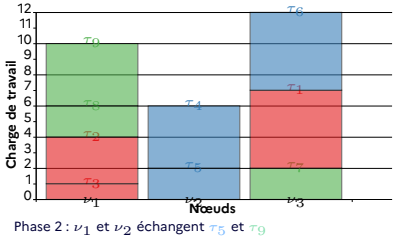


Négociation en deux phases : exemple





Phase 1 : ν_3 délègue τ_3 à ν_1

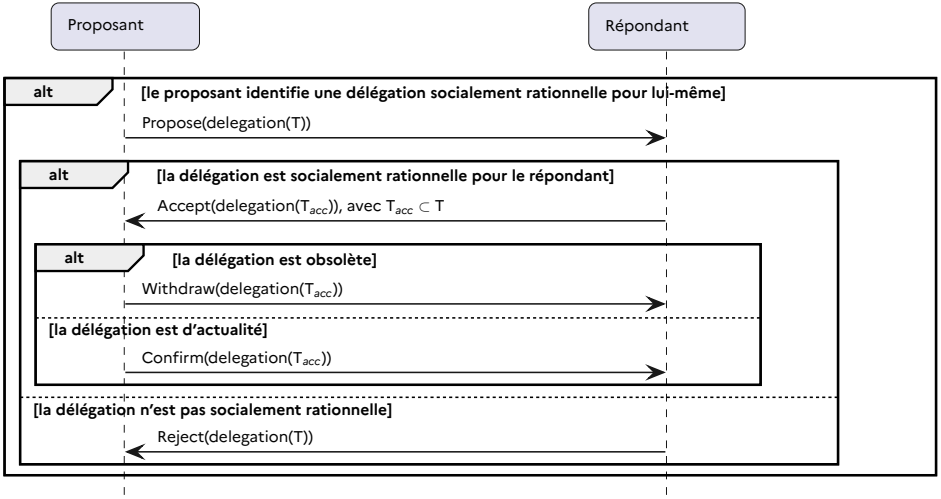


Phase 2 : ν_1 et ν_2 échangent τ_5 et τ_9

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8	τ_9
ν_1		✓	✓					✓	✓
ν_2				✓	✓				
ν_3	✓					✓	✓		

Protocole de négociation

Phase 1



Stratégie de négociation

Phase 1

Donneur : stratégie d'offre

1. **Sélection d'un job :** le job pour lequel le donneur est limitant
2. **Sélection d'un receveur :** le nœud le « moins limitant »
3. **Sélection d'une tâche :** la tâche dont la délégation réduit le plus le coût
4. **Validation :** le donneur croit que la délégation est socialement rationnelle vis-à-vis du *flowtime*

Receveur : règle d'acceptabilité

Le receveur accepte la délégation s'il croit que la délégation est socialement rationnelle vis-à-vis du *flowtime*.

Stratégie de négociation

Phase 2

Donneur : stratégie d'offre

- Stratégie libérale proposant des délégations même si elles ne sont pas acceptables
- Heuristique qui sélectionne les tâches distantes les plus prioritaires

Receveur : stratégie de contre-offre

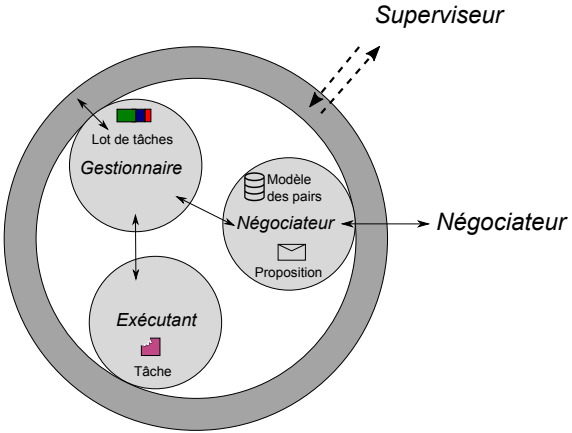
1. **Sélection d'une tâche :** la tâche la plus prioritaire parmi les tâches distantes
2. **Validation :** le receveur croit que l'échange est acceptable

Donneur : règle d'acceptabilité

Le donneur accepte la contre-offre s'il croit que l'échange est acceptable

Une architecture modulaire

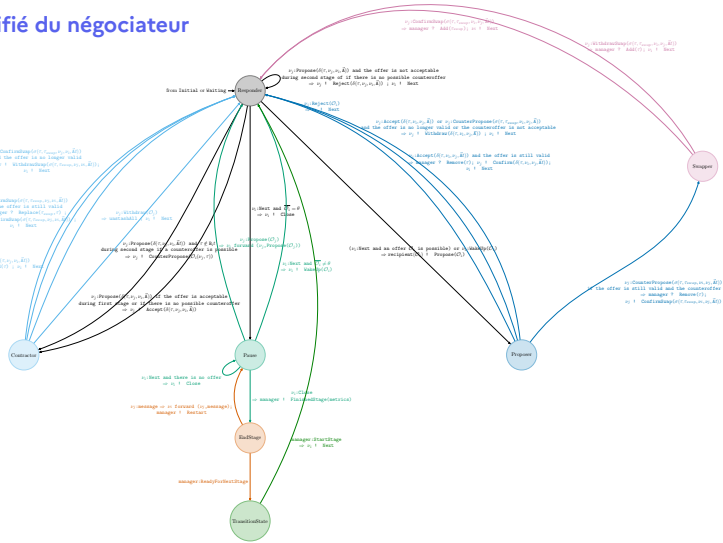
Négociations et consommations concurrentes



- L'**exécutant** consomme les tâches
- Le **négociateur** met à jour une base de croyances pour négocier les tâches avec ses pairs
- Le **gestionnaire** gère le lot de tâches du nœud, ordonnance les tâches à donner à l'exécutant, et retire ou ajoute des tâches selon les négociations effectuées

Une architecture modulaire

Comportement simplifié du négociateur

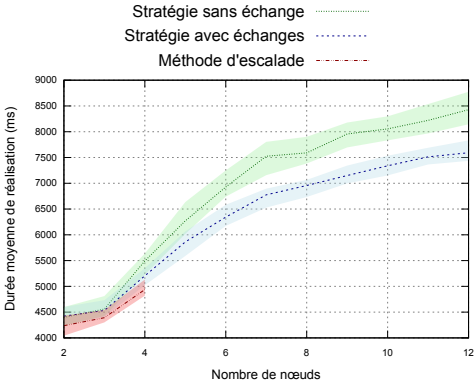


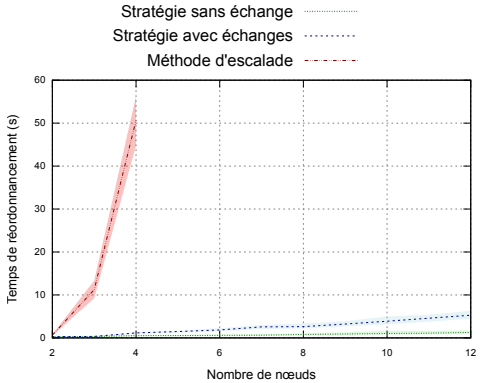
Expérimentations

- Réallocation instantanée
  - La stratégie de contre-offre prolonge la stratégie de délégation et permet d'obtenir un meilleur flowtime au prix d'un surcoût raisonnable
  - Notre méthode permet d'obtenir un meilleur temps d'ordonnancement que les méthodes d'enchères classiques
- Réallocation continue
  - La stratégie de réallocation permet d'améliorer le flowtime
  - Elle est robuste aux aléas d'exécution
  - Elle s'adapte à la libération de nouveaux jobs

Stratégie de contre-offre

Les échanges permettent d'échapper à des minima locaux

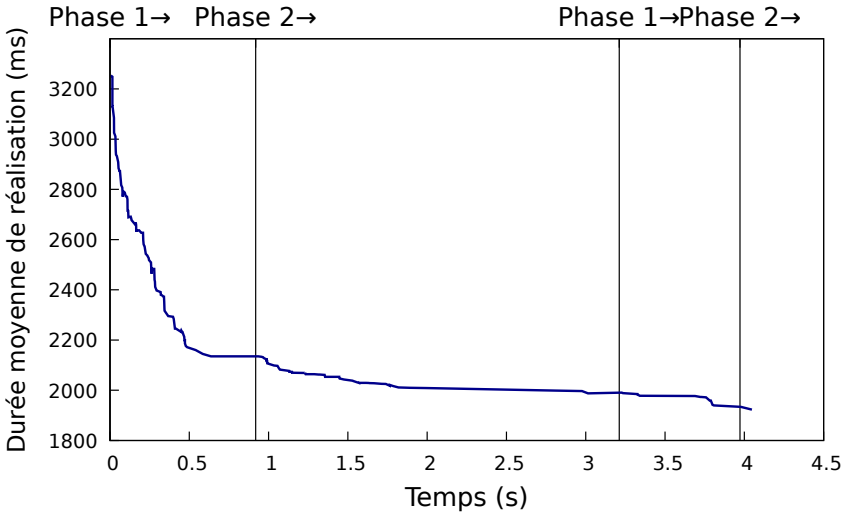




Le *flowtime* obtenu grâce à la stratégie d'échange est meilleur qu'avec la stratégie de délégation seule, tout en gardant un temps de réordonnancement bien inférieur à celui de la méthode d'escalade.

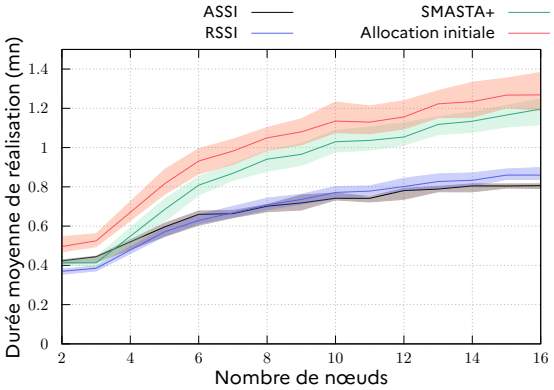
Stratégie de contre-offre

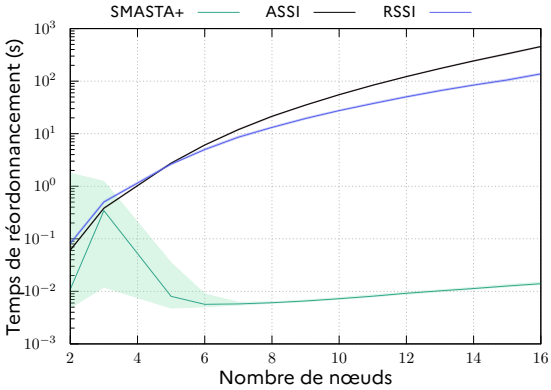
La stratégie d'échange prolonge la stratégie de délégation



Comparaison avec méthodes d'enchères de référence

La stratégie de réallocation permet d'obtenir un meilleur temps de réordonnancement



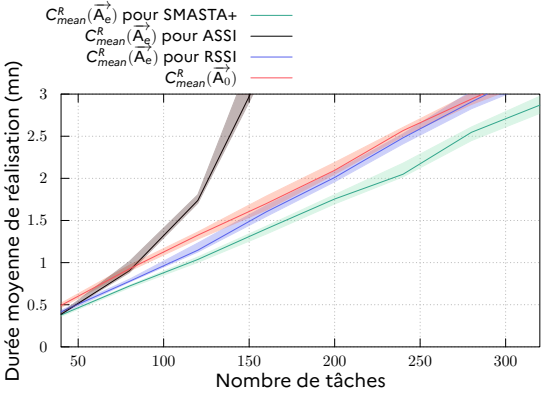


Réallocation continue

La stratégie de réallocation améliore le *flowtime*

Taux d'amélioration de performance :

- Avec RSSI : entre 0 et 15%
- Avec SMASTA+ : entre 13 et 23%

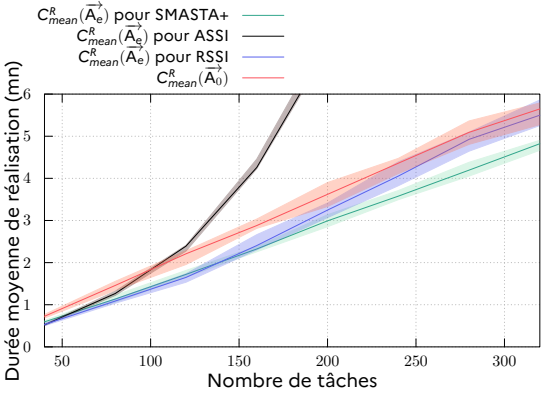


Réallocation continue

La stratégie de réallocation est robuste face aux aléas d'exécution

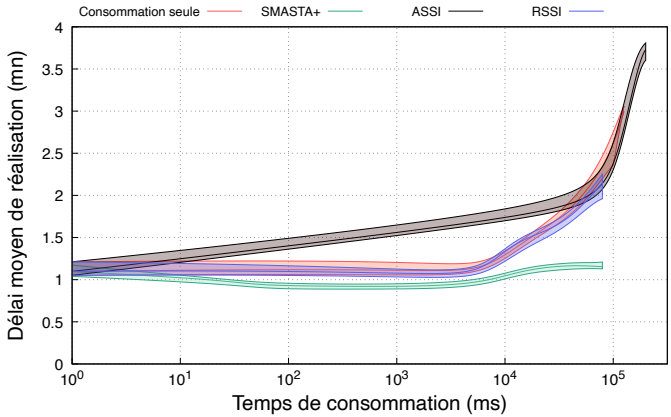
Taux d'amélioration de performance :

- Avec RSSI : entre 0 et 15%
- Avec SMASTA+ : entre 17 et 29%



Réallocation continue

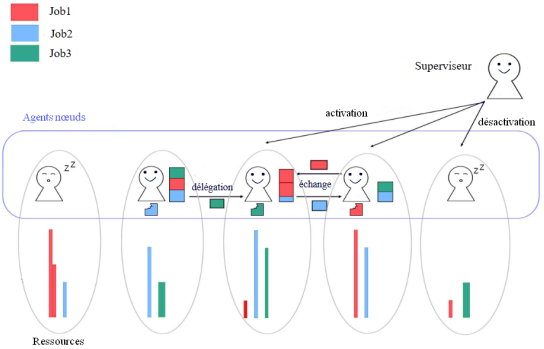
La stratégie de réallocation s'adapte à la libération de nouveaux jobs



À emporter

- **Approche.**
  Stratégie multi-agents pour la réallocation de tâches sur des nœuds réalisée simultanément à la consommation afin de réduire la durée moyenne de réalisation de jobs concurrents
- **Contributions.**
  - Formalisation du problème d'allocation continue de tâches situées
  - Modèle multi-agents où les agents détectent les opportunités de réallocation pertinentes en s'appuyant sur leur stratégie de négociation et leur modèle des pairs
  - Architecture multi-agent modulaire qui permet la concurrence de la négociation et de la consommation des tâches avec 3 agents composants
  - Validation empirique de l'adaptation aux aléas d'exécution et à la libération de jobs
- **Limites.**
  - Hypothèses fortes qui découlent de l'application pratique
  - Notion de localité « binaire »pour les ressources qui ne quantifie pas la distance
- **Travaux futurs.**
  Un processus d'approvisionnement qui permet d'ajouter ou de retirer des nœuds de calcul au cours de l'exécution

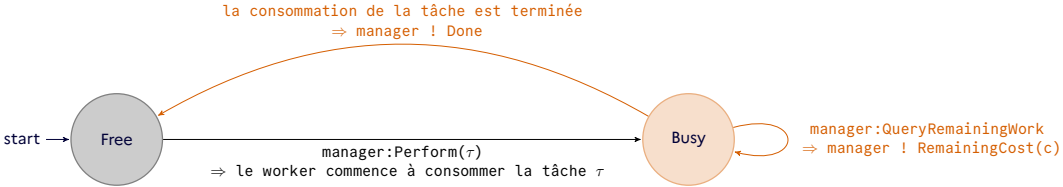
Système multi-agents élastique



Une architecture modulaire

Mise en œuvre des comportements

- Comportement représenté sous la forme d'automate à états finis
- Prototype SMASTA+ implémenté en Scala avec la bibliothèque Akka

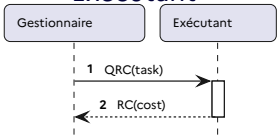


Agent	Nombre d'états	Nombre de transitions	Nombre de lignes
Exécutant	2	3	173
Gestionnaire	5	23	465
Négociateur	9	74	1306
Superviseur	9	69	626

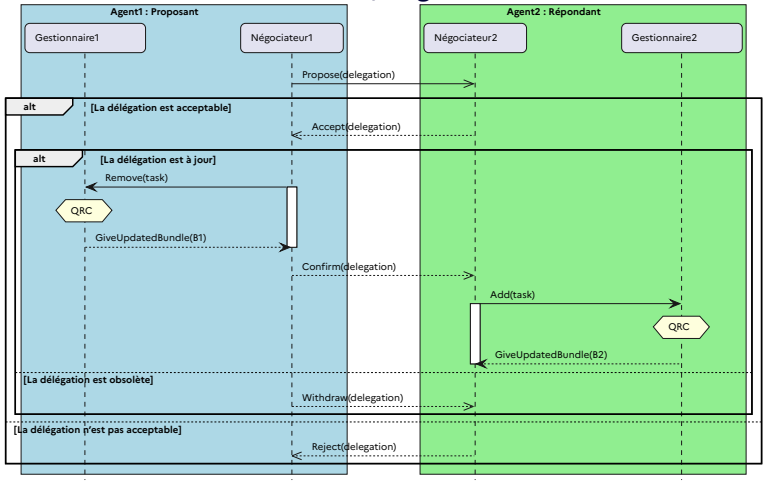
Une architecture modulaire

Interactions entre le gestionnaire, l'exécutant et le négociateur

Gestionnaire/ Exécutant



Gestionnaire/Négociateur



Bibliographie

Baert, Q. (2019).
Négociation multi-agents pour la réallocation dynamique de tâches et application au pa
[thèse de doct.] [Thèse de doctorat dirigée par Routier, Jean-Christophe Caron, Anne-Cécile et Morge, Maxime].

Baert, Q., Caron, A.-C., Morge, M., Routier, J.-C., & Stathis, K. (2021).An adaptive multi-agent system for task reallocation in a MapReduce job. Journal of Parallel and Distributed Computing, 153, 75-88.

Beauprez, E., Caron, A.-C., Morge, M., & Routier, J.-C. (2022). Task Bundle Delegation for Reducing the Flowtime. In ICAART 2021, Revised Selected Papers (p. 22-45, T. 13251). Springer International Publishing.

Beauprez, E., & Morge, M. (2020). Scala implementation of the Extended Multi-agents Situated Task Allocation.

Chen, Y., Mao, X., Hou, F., Wang, Q., & Yang, S. (2016).Combining re-allocating and re-scheduling for dynamic multi-robot task allocation. Proc. of SMC, 395-400.

Choi, H.-L., Brunet, L., & How, J. P. (2009).Consensus-based decentralized auctions for robust task allocation. IEEE transactions on robotics, 25(4), 912-926.