# L'importance des tirages communs pour le consensus probabiliste

Matthieu Perrin et Achour Mostéfaoui
matthieu.perrin@univ-nantes.fr
achour.mostefaoui@univ-nantes.fr

Équipe GDD
LS2N – Université de Nantes

**Keywords :** *Algorithmes probabilistes, Calcul distribué, Consensus*

## Contexte

Le *consensus* est un problème central en algorithmique distribuée, dans lequel des processus cherchent à se mettre d'accord sur une valeur commune parmi celles proposées par chacun d'entre eux. Dans ce stage, nous ferons l'hypothèse que seules les valeurs 0 et 1 peuvent être proposées. Sous cette hypothèse si tous les processus proposent la même valeur, ils doivent garder cette valeur ; sinon, ils doivent seulement s'accorder sur la valeur décidée. L'importance de ce problème vient du fait qu'il a été identifié comme universel pour l'implémentation des objets partagés : si l'on possède une solution au consensus, il est possible d'implémenter tous les objets possédant une spécification séquentielle [1]. Hélas, il n'existe pas d'algorithme *déterministe* implémentant le consensus dans un système asynchrone (les processus ne s'exécutent pas à la même vitesse) et capable de résister aux pannes, ne serait-ce que d'un seul processus [2].

Ce théorème d'impossibilité ne s'applique cependant pas aux algorithmes probabilistes. Ceux-ci utilisent des variables aléatoires et doivent seulement terminer avec probabilité 1. Ils peuvent tolérer jusqu'à $t < \frac{n}{2}$ pannes franches (ou $t < \frac{n}{3}$ attaquants malicieux) pour $n$ processus. On s'intéresse à l'espérance de leur complexité en temps. Une première famille de solutions contient des protocoles utilisant des tirages locaux (*local coins*) : les variables aléatoires tirées par des processus différents sont indépendantes. L'algorithme de Ben-Or [4] (qui a reçu le prestigieux prix Dijskstra en 2016 pour cette idée) fait partie de cette famille, mais nécessite $\mathcal{O}(2^n)$ étapes de communication en moyenne.

Une autre famille d'algorithmes probabilistes utilise des tirages communs (*common coins*) : tous les processus tirent la même séquence de variables aléatoires. Ceci permet d'assurer une terminaison en un nombre moyen de tirages constant [3]. Le problème est que les common coins sont très difficiles à implé-

menter : ils nécessitent plusieurs étapes de communication supplémentaires ou des hypothèses fortes sur l'asynchronie des canaux de communication.

## Objectifs du stage

Après une phase de familiarisation avec le sujet et les problèmes qu'il soulève, le stagiaire devra choisir une question ouverte et tenter d'y répondre, soit en proposant un algorithme, soit en démontrant un théorème d'impossibilité. Nous suggérons d'approfondir la question suivante : le common coin est-il la meilleure abstraction pour implémenter le consensus efficacement ? Parmi les pistes de réflexion, le stagiaire pourra étudier l'implémentation de nouveaux tirages « exotiques » que nous avons identifiés, ou encore chercher une implémentation du consensus qui brise les bornes de complexité démontrées pour le common coin. D'autres questions ouvertes concernent d'adaptabilité du consensus à la puissance de l'adversaire ou au nombre de participants.

Ce stage exige une capacité de raisonnement abstrait avancée, et une certaine familiarité avec la théorie des probabilités. Des connaissances en algorithmique distribuée sont bienvenues, mais pas obligatoires en raison de l'expertise de notre équipe dans ce domaine.

Le stage se déroulera au sein du Laboratoire des Sciences du Numérique de Nantes (LS2N) de l'Université de Nantes. L'équipe GDD (Gestion des Données Distribuées) est reconnue internationalement dans les domaines du calcul distribué et parallèle, du Web et des bases de données.

N'hésitez pas à nous contacter pour en discuter !

## Références


[1] M. Herlihy. *Wait-free synchronization.* ACM Transactions on Programming Languages and Systems (1991)

[2] M. J. Fischer, N. A. Lynch, and M. S. Paterson. *Impossibility of distributed consensus with one faulty process.* Journal of the ACM (1985)

[3] A. Mostéfaoui, H. Moumen, M. Raynal. *Signature-Free Asynchronous Binary Byzantine Consensus with* $t < n/3$*,* $\mathcal{O}(n^2)$ *Messages, and* $\mathcal{O}(1)$ *Expected Time.* Journal of the ACM (2015)

[4] M. Ben-Or, *Another advantage of free choice : completely asynchronous agreement protocols.* ACM Symposium on Principles of Distributed Computing (1983)