



# Tirage Au Sort Distribué

Elèves:

Michaud Thomas  
Sustaita Victor Hugo

Encadrant:

Dauce Emmanuel



## PRÉSENTATION

Notre sujet consiste à faire communiquer  $n$  PCs entre eux ,en s'affranchissant du système classique Serveur/Client, pour qu'ils arrivent à un concensus en un temps fini. C'est-à-dire que chaque PC va communiquer avec les  $(n-1)$  autres et après un temps fini tous seront d'accord,dans notre cas sur une même suite d'entiers.

Le concept de programmation distribuée voit le jour en France dans les années 70 lorsque plusieurs centres universitaires tentent de mettre en commun leurs ressources informatiques. Ses principaux avantages sont les suivant :

- **Augmentation de la puissance de calcul** : on parallélise les calculs en les effectuant sur des machines séparées plutôt qu'une unique machine fasse linéairement tous les calculs
- **Fiabilité** : grâce à la redondance intrinsèque de cette technique l'information est naturellement dupliquée et donc plus fiable
- **Optimisation de l'utilisation des ressources**

## NOTATIONS

**nb\_al**: nombre aléatoire généré par chaque PC (donc différent pour chaque PC)  
**nb\_al[n]**: somme sur les PCs des nb\_al modulo  $n$  (sensé être identique pour tous les PCs)

## ALGORITHME

Voir code en Annexe

## BUGS RECENSÉS

- Lors des cycles un socket client tente d'envoyer un message, après la vérification et l'envoi des nb\_al[k], via un port client non ouvert.

## OPTIMISATIONS SUGGÉRÉES

- Temps de pauses des différents Threads (**temporiser\_ms(int temps)** ). On les retrouve dans fichier: [ligne] :
  - main : [52 ; 87 ; 164]
  - Client :[33]
- Temps des cycles : optimiser le temps de cycle pour qu'il soit assez grand pour que tous les PCs aient le temps de finir les opérations avant d'entammer un nouveau cycle et proche de la limite théorique (due au temps de calcul des opérations et des pauses des Threads)
- System de gestion des clients : On a pensé à faire une liste\_globale contenant tous les clients potentiels (adresses IP) et de faire, en début de cycle, un ping pour vérifier qui est présent et les stocker dans la liste\_adresses\_IP utilisée. Cette technique présente l'avantage de ne pas envoyé le nb\_al du PC à des adresses sans savoir qui écoute.
- Système d'identification (avec mdp?)
- Algorithme du Phase King

## DOCUMENTATION

### VARIABLES

**duree\_cycle:** [int] fixe la durée d'un cycle (envoi + vérification + ajout dans la liste\_finale)  
**numero\_port:** [int] numéro de port utilisé par le serveur (le même pour tous les PCs)  
**taille\_max\_liste\_finale:** [int] paramètre de control sur le nombre d'itération du cycle  
**modulo:** [int] contient le nb\_al[n] le nombre de PC mis en jeu  
**verification:** [boolean] autorise un client à communiquer ou non  
**checkModulo:** [boolean] indique la phase de vérification du nb\_al[n]

### METHODES

**verificationModulo(String fichier):** [boolean] renvoie "true" si tous les PCs renvoie le même nb\_al[n], "false" sinon.  
**date(int annee, int mois, int jour, int heure, int minute, int seconde):** [Date] renvoie un objet Date contenant la date indiquée en arguments.  
**ecriture\_fichier(int nombre, String fichier):** [void] écrit le nombre dans le fichier  
**calcul\_modulo():** [int] renvoie la somme des données de "liste\_temporaire\_reception.txt" modulo le nombre actuel de participants  
**taille\_liste\_adresses\_IP():** [void] initialise taille\_liste\_adresse\_IP (nombre actuel de participants)  
**envoyer\_IP(String fichier):** [void] envoie le nb\_al du serveur aux clients sur fichier

**generation\_nb\_aleatoire(int range): [int]** renvoie un nombre aléatoire entre 1 et range (noté nb\_al)

**temporiser\_ms(int temps): [void]** met le Thread courant en pause pour temps (en ms)

**isValid(): [boolean]** renvoie "true" si le client est sur la liste d'autorisée "false" sinon

## FICHIERS

liste\_temporaire\_modulo.txt: (main) contient la liste des nb\_al[n] au cours d'un cycle

liste\_temporaire\_reception.txt: (main) contient la liste des nb\_al au cours d'un cycle

liste\_finale: (main) contient la liste des nb\_al[n] validé par la vérification