

(Texte public)

Résumé : On analyse un problème issu de l'informatique consistant à répartir de manière équitable un grand nombre de tâches entre deux processeurs parallèles.

Mathématiquement, il s'agit de répartir une suite de variables aléatoires X_1, \dots, X_N indépendantes et uniformément distribuées en deux sous-ensembles disjoints de telle manière que les sommes de ces variables calculées sur les deux sous ensembles soient les plus proches possible.

Mots clefs : Convergence en loi, théorème de la limite centrale, fonction caractéristique.

- *Il est rappelé que le jury n'exige pas une compréhension exhaustive du texte. Vous êtes laissé(e) libre d'organiser votre discussion comme vous l'entendez. Des suggestions de développement, largement indépendantes les unes des autres, vous sont proposées en fin de texte. Vous n'êtes pas tenu(e) de les suivre. Il vous est conseillé de mettre en lumière vos connaissances à partir du fil conducteur constitué par le texte. Le jury appréciera que la discussion soit accompagnée d'exemples traités sur ordinateur.*

Un problème important en informatique est de répartir un grand nombre de tâches entre des processeurs parallèles de manière équitable, c'est-à-dire d'équilibrer leur charges. Nous allons étudier ce problème sur l'exemple d'un modèle élémentaire décrit ci-dessous.

Soient X_1, \dots, X_N des variables aléatoires indépendantes, distribuées uniformément sur $[0, 1]$. La variable X_k correspond au temps d'exécution de la k -ième tâche à effectuer par le processeur. Un partage de tâches sera représenté par une suite d'entiers a_1, \dots, a_N à valeurs dans $\{-1, +1\}$ tel que $a_k = 1$ si la tâche d'indice k est affectée au processeur numéro 1 et $a_k = -1$ si elle est affectée au processeur numéro 2. La différence de charge entre les deux processeurs est alors donnée en valeur absolue par la quantité $|D_N|$ où

$$D_N = \sum_{k=1}^N a_k X_k.$$

1. Répartitions séquentielles

On étudie d'abord deux méthodes de répartition des tâches qui considèrent séquentiellement les tâches X_1, X_2, \dots . On suppose ici que l'on ne connaît pas forcément à l'avance le nombre de tâches à traiter, celles-ci étant affectées à un processeur au fur et à mesure de leur arrivée.

1.1. Répartition au hasard

Essayons tout d'abord d'attribuer les tâches *au hasard* : la tâche d'indice $i \in \{1, 2, \dots, N\}$ est affectée à l'un des deux processeurs avec la probabilité $1/2$, indépendamment des autres indices. Dans ce cas, a_1, \dots, a_N sont des variables aléatoires indépendantes (et aussi indépendantes de X_1, \dots, X_N) identiquement distribuées avec $P(a_1 = 1) = P(a_1 = -1) = 1/2$. Lorsque N tend vers $+\infty$, la suite $(|D_N|/\sqrt{N})_{N \geq 1}$ converge en loi vers une variable aléatoire $|Y|$, où Y suit la loi Gaussienne de moyenne nulle et de variance $1/3$: par conséquent, la politique de répartition *au hasard* n'est pas satisfaisante.

1.2. Répartition markovienne

On décide maintenant d'affecter chaque tâche au processeur qui est le moins chargé. À cette fin, on construit itérativement la séquence a_1, a_2, \dots , de la façon suivante :

- a_1 est une variable aléatoire indépendante de X_1 , telle que $\mathbb{P}(a_1 = +1) = \mathbb{P}(a_1 = -1) = 1/2$, ce qui revient à dire que la première tâche est affectée de façon aléatoire à un des deux processeurs.
- Si a_1, \dots, a_{k-1} sont construits, on pose $D_{k-1} = a_1 X_1 + \dots + a_{k-1} X_{k-1}$. Si $D_{k-1} > 0$, alors le premier processeur est plus chargé que le deuxième et on prend donc $a_k = -1$. Sinon, on pose $a_k = +1$.

On garantit ainsi que D_N est à valeurs dans $[-1, 1]$, ce qui est déjà mieux que la répartition aléatoire et on a par ailleurs la proposition suivante.

Proposition 1.1. *La variable aléatoire D_1 suit la loi uniforme sur $[-1, 1]$ et pour tout $N \geq 2$, la loi de D_N admet pour densité la fonction f qui s'annule en dehors de $[-1, 1]$ et qui vaut $f(x) = 1 - |x|$ pour $x \in [-1, 1]$.*

Indication de preuve. Le cas $N = 1$ est facile et ensuite on procède par récurrence. ■

2. Recherche de répartitions parfaites

Comme les X_k sont des variables à densité, la probabilité qu'il existe une répartition idéale, i.e. vérifiant $a_1 X_1 + \dots + a_N X_N = 0$, est nulle. On dit toutefois que la répartition (a_1, \dots, a_N) est *équitable à ε près* lorsque

$$\left| \sum_{k=1}^N a_k X_k \right| \leq \varepsilon.$$

On démontre alors le résultat suivant.

Proposition 2.1. *Pour tout $\varepsilon > 0$, la probabilité qu'il existe une répartition équitable à ε près, tend vers 1 lorsque $N \rightarrow +\infty$.*

Pour démontrer le résultat, on va exposer un nouvel algorithme de répartition des tâches qui agit sur les tâches d'une façon globale. On démontre en fait le résultat plus fort suivant.

Théorème 2.2. Pour toute suite de tâches $(X_n)_{n \geq 1}$, il existe une suite de répartitions aléatoires notées $(a_1^{(N)}, \dots, a_N^{(N)})$ pour $N \geq 1$, telle que pour tout $N \geq 1$ et tout $k \in \{1, \dots, N\}$ $a_k^{(N)}$ est mesurable par rapport à la tribu engendrée par X_1, \dots, X_N , et pour tout $\varepsilon > 0$ on a :

$$\lim_{N \rightarrow +\infty} \mathbb{P} \left(\left| \sum_{k=1}^N a_k^{(N)} X_k \right| \leq \varepsilon \right) = 1.$$

Indications de preuve. Soient N tâches X_1, \dots, X_N , variables aléatoires indépendantes de densité uniforme sur $[0, 1]$. On construit des variables aléatoires $Y_1^{(N)}, \dots, Y_N^{(N)}$ qui prennent les mêmes valeurs que X_1, \dots, X_N , mais sont triées. Plus précisément, on a

$$\{Y_1^{(N)}, \dots, Y_N^{(N)}\} = \{X_1, \dots, X_N\} \text{ et } Y_1^{(N)} \geq Y_2^{(N)} \geq \dots \geq Y_N^{(N)}.$$

On utilise la même méthode que dans la section précédente, mais en partant des $Y_k^{(N)}$ au lieu des X_k .

- On pose $b_1^{(N)} = 1$.
- Si $b_1^{(N)}, \dots, b_{k-1}^{(N)}$ sont construits, on définit

$$\Delta_{k-1}^{(N)} = \sum_{j=1}^{k-1} b_j^{(N)} Y_j^{(N)}.$$

Si $\Delta_{k-1}^{(N)} > 0$, on pose $b_k^{(N)} = -1$; sinon, on pose $b_k^{(N)} = +1$.

Soit $s(N)$ le plus grand entier de $\{1, \dots, N\}$ tel que $b_{s(N)}^{(N)} = +1$. On vérifie que

$$|\Delta_N^{(N)}| \leq Y_{s(N)}^{(N)}.$$

Par ailleurs, si au moins $\lceil 2/\varepsilon \rceil + 1$ indices k inférieurs à N vérifient $\varepsilon/2 \leq X_k \leq \varepsilon$, alors on vérifie que $Y_{s(N)}^{(N)} \leq \varepsilon$. La probabilité de cet événement tend vers 1 lorsque N tend vers l'infini, on en déduit donc le Théorème 2.2. ■

3. Un théorème limite

La question naturelle qui se pose est alors de préciser quel ordre de grandeur on peut attendre pour une répartition optimale. On pose $\Sigma = \{-1, +1\}^N$ et

$$d_N^* = \inf_{(a_1, \dots, a_N) \in \Sigma} \left\{ \left| \sum_{k=1}^N a_k X_k \right| \right\}.$$

On a le théorème suivant.

Théorème 3.1. Lorsque $N \rightarrow \infty$, on a la convergence en loi

$$(3.1) \quad 2^N \sqrt{\frac{6}{\pi N}} d_N^* \rightarrow W_1,$$

où W_1 est une variable aléatoire de loi exponentielle de paramètre 1.

Démonstration partielle du théorème. Soit $\sigma = (a_1, \dots, a_N)$ un élément de Σ et

$$d_N(\sigma) = \left| \sum_{k=1}^N a_k X_k \right|.$$

Fixons $c > 0$ et soit A_σ l'événement

$$A_\sigma = \left\{ d_N(\sigma) \leq c 2^{-N} \sqrt{\pi N / 6} \right\}.$$

Le théorème est la conséquence directe des lemmes 3.2 et 3.3 ci-dessous.

Lemme 3.2 (Inégalités de Bonferroni). *Soit une suite quelconque d'événements O_1, O_2, \dots, O_m et un entier n tel que $2n + 1 \leq m$. En posant*

$$S_k^{(m)} = \sum_{1 \leq j_1 < j_2 < \dots < j_k \leq m} \mathbb{P}(O_{j_1} \cap O_{j_2} \cap \dots \cap O_{j_k}),$$

on a les inégalités

$$\sum_{k=1}^{2n} (-1)^{k-1} S_k^{(m)} \leq \mathbb{P}\left(\bigcup_{1 \leq i \leq m} O_i \right) \leq \sum_{k=1}^{2n+1} (-1)^{k-1} S_k^{(m)}.$$

Indications de preuve du lemme 3.2. Les inégalités de Bonferroni découlent de l'égalité

$$\mathbb{P}\left(\bigcup_{1 \leq i \leq m} O_i \right) = \sum_{k=1}^{k=m} (-1)^{k-1} S_k^{(m)}.$$

On a en particulier

$$\sum_{\substack{J \subset \Sigma \\ 1 \leq |J| \leq 2n}} (-1)^{|J|-1} \mathbb{P}\left(\bigcap_{\sigma \in J} A_\sigma \right) \leq \mathbb{P}\left(\bigcup_{\sigma \in \Sigma} A_\sigma \right) \leq \sum_{\substack{J \subset \Sigma \\ 1 \leq |J| \leq 2n+1}} (-1)^{|J|-1} \mathbb{P}\left(\bigcap_{\sigma \in J} A_\sigma \right),$$

où l'indice $J \subset \Sigma$ dans une somme indique que la somme est prise sur tous les sous-ensembles J de Σ .

Lemme 3.3. *Soit $k \geq 1$ un entier et notons*

$$\alpha_k(N) = \sum_{\substack{J \subset \Sigma \\ |J|=k}} \mathbb{P}\left(\bigcap_{\sigma \in J} A_\sigma \right).$$

Alors $\alpha_k(N)$ tend vers $c^k / k!$ lorsque N tend vers $+\infty$.

Preuve du lemme 3.3 pour $k = 1$. Il s'agit alors de montrer que, lorsque $N \rightarrow \infty$,

$$\sum_{\sigma \in \Sigma} \mathbb{P}\left(d_N(\sigma) \leq c 2^{-N} \sqrt{\frac{\pi N}{6}} \right) \rightarrow c.$$

Or la somme de ces probabilités est exactement égale à

$$u_N = 2^N \mathbb{P}\left(\frac{|\sum_{k=1}^N V_k|}{\sqrt{N}} \leq c 2^{-N} \sqrt{\frac{\pi}{6}} \right),$$

où les V_k sont des variables indépendantes de loi uniforme sur $[-1, 1]$. En utilisant la transformée de Fourier, on a

$$u_N = \frac{2^N}{2\pi} \int_{-c2^{-N}\sqrt{\pi/6}}^{c2^{-N}\sqrt{\pi/6}} \left(\int_{-\infty}^{+\infty} \exp(-itx) \Phi_N(t) dt \right) dx,$$

où $\Phi_N(t)$ est la fonction caractéristique de $(V_1 + \dots + V_N)/\sqrt{N}$: elle vaut

$$\Phi_N(t) = \left[\operatorname{sinc}\left(\frac{t}{\sqrt{N}}\right) \right]^N, \text{ où } \operatorname{sinc}(u) = \frac{\sin u}{u}$$

est le sinus cardinal. Comme $\Phi_N(t)$ converge vers $\exp(-t^2/6)$, on en déduit que

$$u_N \sim \frac{2^N}{2\pi} \int_{-c2^{-N}\sqrt{\pi/6}}^{c2^{-N}\sqrt{\pi/6}} \left(\int_{-\infty}^{+\infty} \exp\left(-itx - \frac{t^2}{6}\right) dt \right) dx.$$

On conclut en observant que

$$\int_{-\infty}^{+\infty} \exp\left(-itx - \frac{t^2}{6}\right) dt = \sqrt{6\pi} \exp(-3x^2/2),$$

et qu'ainsi u_N tend vers c , ce qu'il fallait démontrer. ■

Suggestions pour le développement

► *Soulignons qu'il s'agit d'un menu à la carte et que vous pouvez choisir d'étudier certains points, pas tous, pas nécessairement dans l'ordre, et de façon plus ou moins fouillée. Vous pouvez aussi vous poser d'autres questions que celles indiquées plus bas. Il est très vivement souhaité que vos investigations comportent une partie traitée sur ordinateur et, si possible, des représentations graphiques de vos résultats.*

— *Modélisation.*

- On pourra s'intéresser à l'efficacité des différents algorithmes de répartition utilisés.
- Pourquoi est-il difficile de trouver la répartition optimale ? On pourra proposer un algorithme qui fournit une bonne répartition approchée.
- Que se passerait-il si on voulait répartir les tâches sur p processeurs ?
- Pourquoi est-il raisonnable d'obtenir un écart optimal de l'ordre de $2^{-N}\sqrt{N}$?

— *Développements mathématiques.*

- On pourra justifier la convergence en loi de la section 1.1, ainsi que la proposition 1.1.
- On pourra détailler la preuve du théorème 2.2.
- Vous pouvez aussi étudier ce qui se passe dans l'algorithme markovien de la section 2 lorsque l'on décide d'affecter systématiquement la première tâche au premier processeur, au lieu de l'affecter à un processeur aléatoire.

- Enfin, vous pouvez compléter la preuve du théorème 3.1 de la section 3 en admettant le lemme 3.2 et les inégalités de Bonferroni, ou bien préciser la preuve de ce lemme, ou encore démontrer les inégalités de Bonferroni.

— *Etude numérique.*

- Vous pouvez illustrer les convergences en loi des répartitions séquentielles.
- Vous pouvez aussi programmer la répartition de la section 2 et vérifier numériquement le théorème 2.2.
- Vous pouvez éventuellement programmer un algorithme qui fournit la répartition optimale, et observer la convergence en loi du théorème 3.1.