File d'attente à processeur partagé selon le marchandage de Nash

Ravaliminoarimalalason T. B.[1] – Andriamanohisoa H. Z.[2] – Randimbindrainibe F.[3]

Laboratoire de Recherche en Sciences Cognitives et Applications (LR-SCA)
Ecole Doctorale en Sciences et Techniques de l'Ingénierie et de l'Innovation
Ecole Supérieure Polytechnique d'Antananarivo
Université d'Antananarivo
BP 1500, Ankatso – Antananarivo 101 – Madagascar

[1] tokybaz@gmail.com - [2] aheryzo@gmail.com - [3] falimanana@mail.ru

Résumé : - Cet article présente une approche par la théorie des jeux de marchandage sur le partage des capacités d'une file d'attente à discipline processeur partagé.
Il présente d'une côté une brève revue d'un tel type de file d'attente et d'autre côté la solution de Nash appliquée à un problème de marchandage. Il démontre la faisabilité du partage insensible et équilibré. Il prouve aussi que le partage n'est pas insensible et équilibré si on intègre la notion de priorité dans la file.

***Mots clés* : -** Théorie des jeux, Nash, Marchandage, File d'attente, Processeur partagé

***Abstract* : -** This paper presents an approach through the bargaining game theory on the capacity sharing in a queue with processor sharing discipline.
It has a brief review of such a type of queue and the Nash bargaining solution. It demonstrates the feasibility of insensitive and balanced sharing. It also proves that sharing is not insensitive and balanced if the notion of priority is integrated in the queue.

***Keywords* : -** Game theory, Nash, Bargaining, Queue, Processor sharing.

1. Introduction :

Le processeur partagé est une limite du Round-Robin lorsque le pas de temps tend vers 0. Il connait un grand succès car il partage toutes les ressources disponibles à travers les clients. Le terme file d'attente est devenu un abus de langage car en réalité il n'y a plus de formation de file, tous les clients qui arrivent sont tout de suite servis avec une partie des ressources du système. [1]Beaucoup d'auteurs ont étudié ce genre de discipline de file d'attente, surtout sa caractéristique insensible par rapport à la loi d'arrivée des clients ou à la loi de demande de service par ces clients. [1] [2] [3] [4]

Ce qui nous intéresse dans cet article c'est plutôt la façon de gérer les ressources à la disposition des clients.

Notre contribution à cette discipline se base sur la théorie des jeux, qui est un domaine très intéressant pour étudier ou même pour prédire le résultat d'une situation conflictuelle comme le partage des ressources de la file à processeur partagé. Nous assimilons le partage à un problème de marchandage qui se voit entre les clients de la file d'attente.

2. File d'attente à processeur partagé

2.1. Description

La discipline de service « Processeur partagé » ou « Processor Sharing » (PS) est une discipline qui partage simultanément la capacité d'un serveur entre tous les clients qui demandent des services à ce serveur.
Une file d'attente adoptant cette discipline pour ordonnancer ses clients est appelée une file d'attente à processeur partagé ou tout simplement une file d'attente PS. Ainsi, pour un partage équitable, à chaque instant t, un client est servi à une vitesse $C/x(t)$ si $x(t)$ clients sont présentent dans une file d'attente PS de capacité C.
Un réseau de files d'attente est dit à processeur partagé si toutes les files d'attente de ce réseau sont des files PS. Un réseau de files d'attente PS peut être donc interprété comme un réseau de files d'attente où chaque file PS i a une capacité $\varphi_i(\mathbf{x})$ qui peut dépendre du nombre de clients $\mathbf{x}$ du réseau.

L'état $\mathbf{x}$ du réseau est un vecteur qui mentionne le nombre de clients dans chaque nœud du réseau : $\mathbf{x} = (x_1, x_2, \dots, x_n)$ où x_i désigne le nombre de client du nœud i.

On appelle donc $\varphi_i(\mathbf{x})$ l'allocation de capacité dans l'état $\mathbf{x}$. On suppose que les services ne puissent être interrompus, c'est à dire qu'on suppose que si $x_i > 0$, alors $\varphi_i(\mathbf{x}) > 0$ pour tout i.

Les arrivées clients au nœud i forment un processus de Poisson d'intensité λ_i. Le temps de service au nœud i de la file i est une variable i.i.d exponentiellement de moyenne $1/\mu_i$.

2.2. Equilibre et insensibilité dans l'allocation de capacité

On considère un réseau de n files d'attente ($n \in \mathbb{N}^*$) dont chaque file i a une capacité $\varphi_i(\mathbf{x})$ qui dépend de l'état $\mathbf{x}$ du réseau. Les capacités d'un tel réseau sont dites équilibrées si :

$$\varphi_i(\mathbf{x})\varphi_j(\mathbf{x} - \mathbf{e_i}) = \varphi_j(\mathbf{x})\varphi_i(\mathbf{x} - \mathbf{e_j}) \qquad (1)$$
$$i, j = 1, \dots, n$$

Où $\mathbf{e_i}$ désigne est un vecteur dont le i-ème composante est 1 et les autres sont toutes nulles.

Cette égalité énonce que le changement sur l'allocation à la file i lorsqu'un client de la file j sort de sa file est identique au changement sur l'allocation à la file j lorsqu'un client de la file i sort aussi de sa file.

Considérant un chemin direct de l'état $\mathbf{x}$ à $\mathbf{0}$, $\langle x, x - e_{i_1}, \dots, x - e_{i_1} - \dots - e_{i_{c-1}}, 0 \rangle$, ce chemin est de longueur c où $c = x_1 + \dots + x_n$ donne le nombre total de clients dans l'état $\mathbf{x}$. On définit une fonction d'équilibre notée Φ la fonction exprimée par :

$$\Phi(\mathbf{x}) = \frac{1}{\varphi_{i_1}(\mathbf{x})\varphi_{i_2}(\mathbf{x} - \mathbf{e_{i_1}}) \dots} \dots \qquad (2)$$
$$\dots \frac{1}{\varphi_{i_c}(\mathbf{x} - \mathbf{e_{i_1}} - \dots - \mathbf{e_{i_{c-1}}})}$$

La propriété d'équilibre impose que la fonction d'équilibre est indépendante du chemin de l'état $\mathbf{x}$ vers l'état $\mathbf{0}$.

Les capacités $\varphi_i(\mathbf{x})$, pour tout i, sont caractérisées par la fonction d'équilibre Φ.

$$\varphi_i(\mathbf{x}) = \frac{\Phi(\mathbf{x} - \mathbf{e_i})}{\Phi(\mathbf{x})} \quad (3)$$

On dit que les capacités sont équilibrées par la fonction Φ.

La distribution stationnaire $\pi(\mathbf{x})$ du nombre de clients dans le réseau peut être exprimée à partir de cette fonction d'équilibre par la relation :

$$\pi(\mathbf{x}) = K\Phi(\mathbf{x}) \prod_{i=1}^{I} \left(\frac{\lambda_i}{\mu_i}\right)^{x_i} \quad (4)$$

Où K est une constante de normalisation d'après la définition probabiliste de la fonction π.

On voit que cette fonction ne dépend pas de la loi d'arrivée des clients ni de leurs demandes de service que par leurs moyennes. Dans ce cas, on dit que le réseau est dit insensible. [1] [3] [5] [6]

3. Jeu de marchandage

3.1. Description d'un jeu de marchandage

Des personnes sont maintenant confrontées à un ensemble d'alternatives tel que leurs préférences sont conflictuelles. Parmi elles, une peut servir de référence qu'on va appeler le statu quo ou point de désaccord, mais bien sûr, il y a d'autres préférences qui peuvent donner plus que le statu quo. Il y a potentiellement un intérêt à ne pas se contenter à entamer à une négociation qui devra conduire à un consensus portant sur une alternative particulière. C'est une alternative qui donne plus à cette personne mais au détriment des autres. Deux cas peuvent se présenter : Soit la négociation aboutit, soit le contraire et dans ce cas, on maintient le statu quo. Ce point joue un rôle de menace pour ces personnes.

Ce genre de jeu s'appelle un jeu de marchandage, qui est une des branches de la théorie des jeux. Nash a décrit une solution qui prédit le résultat d'un tel marchandage.

3.2. Un problème de marchandage

Un problème de marchandage est défini par un couple (U, d) où $d \in U$ et U un sous-ensemble de $\mathbb{R}^n$ satisfaisant les propriétés suivantes :

- U est convexe et fermé
- U est borné :

$\exists \bar{u} \in \mathbb{R}^n$ tel que $u_i < \bar{u}_i$ pour tout u de U

- d est dominé au sens de Pareto :

$\exists u \in U$ tel que $u_i > d_i$ pour tout i de 1 à n

U est non vide car il contient au moins le point de désaccord ou le statu quo. La convexité de U reflète le fait que l'ensemble des alternatives est un continu et que les joueurs n'ont pas un penchant pour le risque. U borné garantit un limite sur le gain des joueurs, et la domination au sens de Pareto invoque qu'un gain potentiel existe vraiment.
Une solution à ce problème marchandage est une règle décrite par une fonction f telle qu'à tout problème de marchandage (U, d) associe l'élément $\bar{u} = f(U, d)$ de l'ensemble U.

$\bar{u}_i = f_i(U, d)$ où f_i définit le niveau d'utilité pour chaque i.

Les solutions de ce marchandage doivent satisfaire à certaines conditions. [7]

3.3. La solution de Nash

La solution de Nash se fonde sur des axiomes qui conduisent à l'unicité de la solution. Celle-ci correspond à la maximisation de la fonction $f(u)$ sur l'ensemble U.

$$f(u) = \prod_{i=1}^{n}(u_i - d_i)^{\omega_i} \qquad (5)$$

ω_i désigne un éventuel poids du joueur i du marchandage.

Cette fonction est continue sur un ensemble compact, donc la solution existe. L'unicité tient au fait que l'ensemble U est convexe et la fonction f est strictement concave, ce qui se traduit par des courbes de contour strictement convexes par rapport à l'origine. [8]

4. Marchandage de ressource d'une file d'attente PS

4.1.Formulation du jeu

On va étudier le cas d'un seul serveur à file à processeur partagé. Les clients de cette file sont divisés en classe de clients numérotée de 1 à n. Cette file est assimilable à n files en parallèle dont chaque file i sert un client de classe i. A un instant quelconque, chaque file i sert x_i clients et l'état du système est le vecteur $\mathbf{x} = (x_1, \dots, x_n)$.

Le système tout entier héberge alors $|\mathbf{x}|$ clients où $|\mathbf{x}|$ désigne la somme de toutes les composantes du vecteur $\mathbf{x}$: $|\mathbf{x}| = \sum x_i$.

La somme des capacités individuelles $\varphi_i(\mathbf{x})$ de chaque file i dépendant de l'état $\mathbf{x}$ du système est donc égale à la capacité totale C disponible pour le système.

$$\sum_{i=1}^{n} \varphi_i(\mathbf{x}) = C \qquad (6)$$

Pour chaque file i, la ressource $\varphi_i(\mathbf{x})$ disponible est partagée équitablement entre les x_i clients. La ressource disponible pour un client de classe i est donc :

$$r_i = \frac{\varphi_i(\mathbf{x})}{x_i} \qquad (7)$$

On va modéliser ce réseau de files d'attente PS à un jeu dont les joueurs sont les clients du réseau qui marchandent les ressources disponibles sur le réseau tout entier.

Chaque file i, a pour fonction d'utilité $f\big(\varphi_i(\mathbf{x})\big)$.

4.2. Solution de ce problème de marchandage

La solution de ce problème de marchandage est :

$$\begin{aligned} &\arg\max_{\varphi_i} \sum_{i=1}^{n} x_i . f(\varphi_i) \\ &\text{sous contrainte : } \sum_{i=1}^{n} \varphi_i = C \end{aligned} \qquad (8)$$

Le terme $x_i . f(\varphi_i)$ désigne que l'utilité sur la file i est commune à tous les x_i clients, et que l'utilité globale se calcule à travers tous ces clients.

Celui-ci ramène à un problème d'optimisation convexe, et la résolution utilise les multiplicateurs de Lagrange avec les conditions de Karush–Kuhn–Tucker (KKT).

La fonction associée pour trouver les multiplicateurs de Lagrange est donnée par :

$$\mathcal{L}(\mathbf{x},\lambda) = \sum_{i=1}^{n} x_i . f(\varphi_i) + \lambda . \left(\sum_{i=1}^{n} \varphi_i - C\right) \quad (9)$$

La fonction d'utilité f peut être choisie pour que la solution obtenue soit égale à la solution de Nash avec l'origine comme point de désaccord. Dans ce cas, il suffit d'utiliser une fonction logarithme :

$$f(\varphi_i) = \omega_i . \log(\varphi_i) \quad (10)$$

Le choix de cette fonction dérive de la fonction efficacité qui donne la proportion de capacité obtenue par la file i par rapport à la capacité totale du système C : $\log(\varphi_i) / \log(C)$. Le terme ω_i désigne un poids de chaque classe de client.
La fonction de Lagrange associée est donc :

$$\mathcal{L}(\mathbf{x},\lambda) = \sum_{i=1}^{n} x_i . \omega_i . \log(\varphi_i) + \lambda . \left(\sum_{i=1}^{n} \varphi_i - C\right) \quad (11)$$

Parmi les conditions de KKT, pour tout k, on a :

$$\frac{x_k . \omega_k}{\varphi_k} + \lambda = 0 \quad (12)$$

Qui donne :

$$\frac{x_i . \omega_i}{\varphi_i} = \frac{x_j . \omega_j}{\varphi_j} \forall\, i \neq j \quad (13)$$

Et $\sum_{i=1}^{n} \varphi_i = C$

La résolution de ce système d'équations montre que la capacité qu'on doit allouer à la file i est :

$$\varphi_i(x) = \frac{\omega_i . x_i}{\sum_{k=1}^{n} \omega_k . x_k} . C \quad (14)$$

A première vue, c'est une sorte d'allocation proportionnelle aux nombres de clients dans chaque file et aux poids attribués à chaque file.

4.3.Equilibre et insensibilité

D'après les capacités allouées à chaque file :

$$\varphi_i(x) = \frac{\omega_i . x_i}{\sum_{k=1}^{n} \omega_k . x_k} . C \quad (15)$$

$$\varphi_j(x) = \frac{\omega_j . x_j}{\sum_{k=1}^{n} \omega_k . x_k} . C \quad (16)$$

$$\varphi_i(x - e_j) = \frac{\omega_i . x_i}{\left(\sum_{k=1}^{n} \omega_k . x_k\right) - \omega_j} . C \quad (17)$$

$$\varphi_j(x - e_i) = \frac{\omega_j . x_j}{\left(\sum_{k=1}^{n} \omega_k . x_k\right) - \omega_i} . C \quad (18)$$

L'équation d'équilibre (1) nous donne :

$$\frac{1}{\left(\sum_{k=1}^{n}\omega_k.x_k\right)-\omega_j} = \frac{1}{\left(\sum_{k=1}^{n}\omega_k.x_k\right)-\omega_i} \quad (19)$$

Cette égalité est vraie si et seulement si $\omega_i = \omega_j$ pour tout $i \neq j$.

C'est à dire qu'on doit attribuer le même poids à toutes les files pour avoir un système équilibré et insensible. En d'autre terme, ceci montre que le fait de doter au système la notion de priorité à partir des poids l'enlève de sa caractéristique d'équilibre et d'insensibilité.

Naturellement, on doit toujours attribuer plus de ressources à la file de classe plus prioritaire et la notion d'équilibre définie dans le deuxième paragraphe ne sera plus maintenue.

4.4. La distribution stationnaire

Dans toute la suite, on va étudier un système équilibrée et insensible, avec une capacité de chaque file égale à :

$$\varphi_i(\mathbf{x}) = \frac{x_i}{\sum_{k=1}^{n}x_k}.C = \frac{x_i}{|\mathbf{x}|}.C \quad (20)$$

On a :

$$\varphi_i(\mathbf{x}) = \frac{\Phi(\mathbf{x}-\mathbf{e_i})}{\Phi(\mathbf{x})} = \frac{x_i}{|\mathbf{x}|}.C \quad (21)$$

Donc :

$$\Phi(\mathbf{x}) = \Phi(\mathbf{x}-\mathbf{e_i}).\frac{|\mathbf{x}|}{C.x_i} \quad (22)$$

$$\Phi(\mathbf{x}-\mathbf{e_i}) = \Phi(\mathbf{x}-2\mathbf{e_i}).\frac{|\mathbf{x}|-1}{C.(x_i-1)} \quad (23)$$

$$\Phi(\mathbf{x}-x_i\mathbf{e_i}) = \Phi(\mathbf{x}-(x_i-1).\mathbf{e_i}).\frac{|\mathbf{x}|-x_i}{C} \quad (24)$$

Ainsi :

$$\Phi(\mathbf{x}) = \Phi(\mathbf{x}-x_i\mathbf{e_i}).\frac{|\mathbf{x}|\ldots(|\mathbf{x}|-x_i)}{C^{x_i}.x_i!} \quad (25)$$

En effectuant la même démarche pour tout i, on obtient :

$$\Phi(\mathbf{x}) = \Phi(\mathbf{O})\frac{|\mathbf{x}|!}{C^{|\mathbf{x}|}.(x_1!\ldots x_n!)} \quad (26)$$

Et la distribution stationnaire en découle :

$$\pi(\mathbf{x}) = K.\frac{|\mathbf{x}|!}{C^{|\mathbf{x}|}.(x_1!\ldots x_n!)}.\prod_{i=1}^{n}\rho_i^{x_i} \quad (27)$$

Où K désigne une constante normalisatrice obtenue à partir de la définition probabiliste de π : $\sum \pi = 1$, et ρ_i l'intensité de trafic sur la file i.

5. Conclusion

L'approche par la théorie des jeux de marchandage abordée dans ce travail nous a permis au partage des ressources d'une file d'attente à processeur partagé.

Cette méthode avec la solution de Nash nous a permis de trouver comment partager ces ressources à travers des classes de clients.

Cette approche nous a aussi permis de démontrer que la notion de priorité dans la file d'attente PS la fait perdre la notion d'insensibilité et d'équilibre en terme de partage de cette ressource.

Références

[1] Ravaliminoarimalalason T. B., *« Etude du réseau de Whittle et son application pour modéliser les réseaux à commutation de paquets »*, DEA Sciences Cognitives et Applications, ESPA - Université d'Antananarivo, 2013

[2] Minh A. T., « *Insensibilité dans les réseaux de files d'attente et applications au partage de ressources informatiques* », Groupe de Recherche sur la théorie des réseaux et communications, Th. Doctorat, 2007

[3] Proutiere A., « *Insensibilité et bornes stochastiques dans les réseaux de files d'attente* », FranceTelecom R&D - Ecole Polytechnique, Th. Doctorat, 2003

[4] Abhay K., Parekh J., « *A generalized processor sharing approach to flow control in integrated services networks* », The Johns Hopkins University and Massachusetts Institute of Technology, Th. Doctorat, 1992

[5] Dao T. T. H., « *Zero-automatic queues and networks* », Laboratoire d'Informatique Algorithmique - Université Paris Diderot, Th. Doctorat, 2007

[6] Ward A. R., « *Predicting response times in processor sharing queues* », Dpt of Management Science and Engineering - Stanford University, AT&T Labs Shannon Laboratory, 2000

[7] Dehez P., *« Conflit, marchandage, partage, pouvoir »*, Core-UCL, 2007

[8] Alexander C. O., Ledermann W., « *The constrained Nash Bargaining solution* », J. Opl Res. Soc. Vol. 45, University of Sussex, 1994