RAIRO Operations Research
RAIRO Oper. Res. **34** (2000) 99-129

TRANSFERT DE CHARGE DANS UN RÉSEAU DE PROCESSEURS TOTALEMENT CONNECTÉS (*)

par Maryse BÉGUIN ([1])

Communiqué par Bernard LEMAIRE

Résumé. – *L'étude présentée ici modélise un transfert de charge sur un réseau de processeurs totalement connectés. Chaque processeur peut accueillir au plus K tâches. Une différence de deux charges entre deux processeurs est une situation interdite, et un transfert immédiat et instantané est déclenché dès que cette situation se produit. Les performances du système sont évaluées par les indices suivants : probabilité de rejet, nombre moyen de tâches traitées par unité de temps, temps de réponse moyen, probabilité stationnaire pour un processeur d'accueillir i tâches. Le but de cette étude est de mesurer les répercussions du transfert de charge en comparant les valeurs des indices obtenues avec transfert avec celles obtenues sans transfert. En particulier, le comportement asymptotique pour des systèmes massivement parallèles est étudié et interprété. Calculées dans une situation idéale, ces comparaisons permettent d'obtenir des bornes supérieures sur les bénéfices que l'on peut attendre d'un réel transfert. Elles permettent également d'étudier l'opportunité du transfert selon les valeurs des paramètres du système. Le nombre moyen de transferts effectués par unité de temps et le nombre moyen de transferts pour une tâche donnée sont calculés. L'asymptotique quand K tend vers l'infini est également étudiée.*

Mots clés : Évaluations de performances, transfert de charge, système massivement parallèle, processus de Markov, processus de naissance et de mort.

Abstract. – *In this paper, a model of the load transfer on a fully connected net is presented. Each processor can accept at most K tasks. A load difference of two tasks between two processors is a prohibited situation and when it may appear, an immediat and instantaneous transfer is decided.*

The performances of the system are evaluated by the following indices: the reject probability, the throughput, the mean response time, the stationary probability distribution for a processor to host i tasks. The aim of this study is to evaluate the load transfer inpact thanks to the comparison between the values of the indices without transfer and with transfer. In particular the asymptotic behaviour for massively parallel systems is studied and interpreted. Calculated with an ideal situation, these comparisons yield upper bounds on the benefits that can be expected from a transferring policy. Beyonds, the opportunity of the transfer according to the values of the parameters can be studied. The mean number of transfers executed within a time unit and the mean number of transfers of a given task are calculated. At last values of the indices when the number of accepted tasks K grows to infinity is studied.

Keywords: Performance evaluation, load transfer, massively parallel system, Markov process, death and birth process.

(*) Reçu en décembre 1997.

([1]) SMS-LMC-IMAG, BP. 53, 38041 Grenoble Cedex, France.

Recherche opérationnelle/Operations Research, 0399-0559/00/01/$ 7.00

1. INTRODUCTION

Le développement relativement récent des multi-calculateurs permet de traiter des problèmes complexes issus de domaines divers et nécessitant de très grosses puissances de calcul. Comme dans les systèmes répartis, se posent des problèmes cruciaux comme l'utilisation optimale des ressources disponibles et plus particulièrement des processeurs. Nous ne connaissons pas encore de données permettant d'estimer la durée de vie et la loi des tâches générées par des machines parallèles, mais par similitude avec les travaux effectués sur les réseaux de stations de travail, il semble indispensable que le système d'exploitation gère l'allocation des tâches sur les différents processeurs afin d'éviter les situations où certains processeurs sont surchargés alors que d'autres sont oisifs (*cf.* [11]). Cette gestion implique une distribution de la charge globale sur les différents processeurs pouvant entraîner des transferts de charge d'un processeur à un autre.

La prédiction du comportement futur de la charge est complexe et justifie une étude stochastique. Cette problématique est ancienne et je rappellerai ici les études récentes de modélisation qui ont été proposées. En 1993, Evans et Butt [5] ont étudié par la théorie des files d'attente et des simulations la probabilité de transfert entre deux sites d'un algorithme d'équilibrage de charge avec pénalisation du transfert. En 1994, Squillante et Nelson [15] ont fait une étude analytique par les matrices stochastiques du temps de réponse moyen pour un transfert de charge basé sur des politiques de seuils. Dans la même année, Malyshev et Robert [9] ont obtenus des valeurs asymptotiques de la probabilité de perte par des processus de Markov et des fonctions de Lyapounov d'algorithmes de transfert de charge global et local pour une infinité de sites de capacité 1. En 1995, Guyennet *et al.* [14] ont comparé des résultats analytiques et des simulations donnant la valeur d'un indice d'efficacité d'un modèle markovien d'équilibrage de charge entre des sites de capacité infinie. Néanmoins le cas des systèmes massivement parallèles de sites de capacité K n'a pas été étudié. Nous proposons ici la modélisation et l'évaluation d'un algorithme de transfert de charge basé sur une évolution markovienne de la configuration des charges des processeurs. L'objectif de cette étude est de comprendre comment opère l'algorithme de balance global sur les indices classiques d'évaluation de performance (*cf.* [3]), et de comparer les valeurs de ces indices avec et sans politique de transfert.

L'article sera organisé de la façon suivante : la description formelle du modèle est donnée dans la section 2. Les méthodes analytiques et les

résultats obtenus pour le calcul des indices retenus pour cette étude sont explicitées dans la section 3. Pour un nombre donné de sites, les évolutions des indices en fonction des paramètres du système sont étudiées en section 4. Le comportement des systèmes massivement parallèles, et la comparaison des valeurs théoriques avec les valeurs obtenues pour un nombre raisonnable de sites dans un réseau sont analysés dans la section 5. Dans la section 6 les bornes supérieures sur le bénéfice que l'on peut attendre d'un réel transfert sont déduites et l'analyse de quelques problèmes d'optimisation est proposée. Une estimation du coût des transferts est étudiée et analysée dans la section 7. La section 8 fournit les résultats obtenus pour cet algorithme lorsque la capacité mémoire de chaque site peut être considéré comme infinie. Enfin la section 9 tire les conclusions et les perspectives qui peuvent être déduites de cette étude.

2. DESCRIPTION DU MODÈLE

Le contexte est le suivant : on dispose de n processeurs qui reçoivent chacun un flot de requêtes gérée grâce à une file d'attente selon la politique FIFO. Le comportement moyen d'une application de grande taille est modélisé, et on suppose pour cela qu'une unité de tâche est définie, et que les contraintes de précédence entre les tâches sont noyées dans l'ensemble des tâches qui sont exécutées par le réseau. La capacité mémoire associée à chaque processeur est de K tâches, et l'ensemble du processeur de sa file d'attente et du contrôleur le reliant au réseau d'interconnection sera désigné par un « site ». Les sites d'un réseau informatique peuvent être considérés comme les sommets d'un graphe, dont les arêtes représentent les connexions physiques entre ces sites.

Les hypothèses de modélisation sont alors les suivantes : dans ce modèle, le graphe est supposé complet, donc chaque processeur communique avec l'ensemble des autres. La durée d'exécution des tâches sur chaque site suit une loi exponentielle de paramètre μ, et le processus d'arrivée des tâches par site est un processus de $\mathcal{P}$oisson de paramètre λ. La charge de chaque site est définie par le nombre de tâches qu'accueille ce site. À chaque instant, ce nombre est un entier entre 0 et K, et une tâche générée par un site de charge K est rejetée.

Deux sites se transfèrent instantanément des tâches dans les deux situations suivantes. Quand une tâche arrive sur un site dont la charge devient j, cette tâche est transférée sur un site de charge $j-2$ s'il existe. Parmi les sites de charge $j-2$, le site qui reçoit la tâche supplémentaire est choisi au hasard.

Quand une tâche se termine sur un site dont la charge devient j, une tâche en provenance d'un site de charge $j+2$ est transférée. Parmi les sites de charge $j+2$, celui qui transfère sa dernière tâche arrivée est choisi au hasard.

Pour que le transfert ait un sens, la contrainte sur la capacité mémoire K est d'être supérieure à 2.

Dans la situation sans transfert, les n sites se comportent comme n files indépendantes $M/M/1/K$ de taux d'arrivée λ, et de taux de service μ (*cf.* [13]). Par comparaison avec les files $M/M/1$, le système sera dit « saturé » lorsque le taux de service sera inférieur au taux d'arrivée.

Dans la situation avec transfert, le processus étudié est le processus $\{X_t, t \geq 0\}$, qui à chaque instant t fait correspondre le n-uplet dont la i-ème coordonnée représente la charge du $i-$ème site. La différence avec une file d'attente $M/M/n/Kn$ (*cf.* [4]) tient au respect du comportement local de chaque site. En effet, dans la modélisation présentée ici, il n'y a pas de file d'attente partagée par l'ensemble des sites et un site de charge K n'accepte plus et ne génère plus de nouvelles tâches. Quand il y a partage de l'espace mémoire, le protocole de transfert instantané de la surcharge sur des sites sous-chargés conduit à un modèle de file d'attente multiserveurs ayant n serveurs et une capacité globale de nK tâches. Ce type de files d'attente ($M/M/n/nK$) s'analyse avec des techniques classiques de traitement des processus aléatoires de naissance et de mort (*cf.* [7]).

Les indices suivants qui apparaissent pertinents pour la plupart des problèmes posés seront étudiés. En effet ces indices répondent à la demande des évaluations de performance pour les machines parallèles et rendent comptent du bon fonctionnement d'un système (*cf.* [3]).

Débit du système. C'est le nombre moyen $\overline{T}$ de tâches traitées par unité de temps par l'ensemble du système (*cf.* [16]).

Saturation mémoire. C'est la probabilité P_{sat} que l'espace mémoire d'un site soit saturé (K tâches présentes) et que, par conséquent, toute nouvelle génération de tâche par l'application sur ce site soit rejetée par le système. Cette probabilité peut être interprétée comme une mesure de la dégradation du système, et cette probabilité doit être minimisée pour garantir une utilisation optimale de l'ensemble des mémoires du système.

Charge de travail. C'est le nombre moyen $\overline{N}$ de tâches présentes sur l'ensemble des sites pendant une unité de temps.

Temps de réponse moyen. C'est le temps moyen $\overline{R}$ écoulé entre l'instant où la tâche est générée par l'application, et l'instant où la tâche est terminée. Du point de vue de l'utilisateur, et en l'absence de toute autre spécification, cet indice doit être minimisé (*cf.* [7]).

Mesure stationnaire de charge. C'est la probabilité P_i pour un site donné d'avoir une charge égale à i. Pour un nombre donné de sites, ces probabilités représentent la proportion de sites ayant une charge égale à i. Ces probabilités ne sont donc pas à proprement parler des indices de performance, mais représentent les lois marginales de la mesure produit sur l'espace des états. Leurs évolutions selon les paramètres du système permettent de rendre compte du comportement global de celui-ci, et permettent de mieux comprendre comment opère le transfert de charge.

3. RÉSULTATS FORMELS AVEC ET SANS TRANSFERT

Le processus $\{X_t, t \geq 0\}$ défini à la section 2 est un processus de Markov admettant une unique mesure stationnaire. Tous les indices explicités dans la section 2 sont calculés en régime stationnaire dans les situations avec et sans transfert. Pour les distinguer, toutes les quantités relatives à la situation sans transfert seront notées avec un astérisque « $*$ ».

3.1. Propriétés structurelles dans les deux situations

Dans un premier temps, nous nous intéressons aux probabilités stationnaires pour un site donné d'avoir une charge i. Compte tenu des rôles symétriques joués par chacun des sites, ces quantités ne dépendent pas du site. Pour chaque niveau de charge i les probabilités stationnaires de chaque site d'avoir une charge i avec et sans transfert respectivement seront notées $P_i(\frac{\lambda}{\mu})$ et $P_i^*(\frac{\lambda}{\mu})$. Dans les deux situations, avec et sans transfert, des équations de nature algébrique relient certaines valeurs des $P_i(\frac{\lambda}{\mu})$.

PROPOSITION 1 : *Les propriétés suivantes sont vérifiées dans les deux situations avec et sans transfert :*

Propriété de symétrie :

$$1) \quad \forall\, i \in \{0 \ldots K\} \quad P_i\left(\frac{\lambda}{\mu}\right) = P_{K-i}\left(\frac{\mu}{\lambda}\right),$$

Propriété d'équilibre :

$$2) \quad \lambda\left(1 - P_K\left(\frac{\lambda}{\mu}\right)\right) = \mu\left(1 - P_0\left(\frac{\lambda}{\mu}\right)\right).$$

Commentaires : la première partie de cette proposition revient à l'observation suivante. Remplacer i par $K-i$ revient à remplacer la capacité mémoire occupée d'un site par sa capacité mémoire libre. Le processus qui à chaque instant t fait correspondre le n-uplet dont la i-ème coordonnée représente la capacité mémoire libre du $i-$ème site a exactement la même dynamique que le processus $\{X_t,\, t \geq 0\}$, en inversant les rôles de λ et μ.

La deuxième partie de la proposition est une équation d'équilibre ou de balance. La partie gauche représente le nombre moyen de tâches acceptées par chaque site par unité de temps, tandis que la partie droite représente le nombre moyen de tâches traitées par chaque site par unité de temps.

Néanmoins, la démonstration de ces résultats avec et sans transfert, anticipe sur les résultats obtenus dans la section 3.3 et la section 3.2 mais ils peuvent être vérifiés par des techniques d'algèbre classique.

La détermination des lois marginales $P_i(\frac{\lambda}{\mu})$ suffit pour calculer les valeurs de tous les indices de performance retenus dans ce modèle grâce aux relations algébriques données dans la proposition 2.

PROPOSITION 2 : *Les indices de performance et les lois marginales* $P_i(\frac{\lambda}{\mu})$ *sont liés par les équations algébriques suivantes :*

$$\begin{aligned}
P_{sat} &= P_K\left(\frac{\lambda}{\mu}\right), \\
\overline{T} &= n\lambda\left(1 - P_K\left(\frac{\lambda}{\mu}\right)\right), \\
\overline{N} &= n\left[\sum_{j=1}^{K} jP_j\left(\frac{\lambda}{\mu}\right)\right], \\
\overline{R} &= \frac{1}{\mu} + \frac{1}{\lambda}\left(\frac{\sum_{j=2}^{K}(j-1)P_j(\frac{\lambda}{\mu})}{(1 - P_K(\frac{\lambda}{\mu}))}\right).
\end{aligned}$$

Commentaire : la dernière équation exprime le temps de réponse moyen d'une tâche comme la somme du temps de service moyen et du temps d'attente moyen.

Démonstration. Les trois premières équations sont aisément obtenues (*cf.* [13]). La dernière formule de la proposition 2 s'obtient en vérifiant que les hypothèses de validité de la formule de Little (*cf.* [8] démontrée par exemple dans [16] p. 100) sont effectivement vérifiées dans ce contexte, et en utilisant la propriété d'équilibre.

La suite de cette section utilise les résultats classiques sur les processus de naissance et de mort pour lesquels une référence est par exemple Barucha-Reid [1]. Pour alléger les notations, et sans perte de généralité, les calculs sont effectués avec un temps moyen d'exécution égal à 1, ce qui revient à fixer l'unité de temps $\frac{1}{\mu}$ à 1.

3.2. Résultats sans transfert

Sans transfert, les n sites se comportent comme n files $M/M/1/K$ indépendantes, d'où les résultats.

PROPOSITION 3 : *Pour le modèle n sites de capacité K de taux d'arrivée λ, les valeurs des indices de performance sans politique de transfert sont les suivantes :*

$$
\begin{aligned}
\forall j \in \{0 \dots K\} \quad P_j^*(\lambda) &= \lambda^j \frac{1-\lambda}{1-\lambda^{K+1}} \\
&\quad si\, \lambda \neq 1, \qquad = \frac{1}{K+1} \quad sinon, \\
P_{\text{sat}}^* &= \lambda^K \frac{1-\lambda}{1-\lambda^{K+1}} \\
&\quad si\, \lambda \neq 1, \qquad = \frac{1}{K+1} \quad sinon, \\
\overline{T}^* &= n\lambda \left(\frac{1-\lambda^K}{1-\lambda^{K+1}} \right) \\
&\quad si\, \lambda \neq 1, \qquad = n\frac{K}{K+1} \quad sinon, \\
\overline{N}^* &= n\lambda \left(\frac{1}{1-\lambda} - \frac{(K+1)\lambda^K}{1-\lambda^{K+1}} \right) \\
&\quad si\, \lambda \neq 1, \qquad = n\frac{K}{2} \quad sinon, \\
\overline{R}^* &= 1 + \frac{\lambda}{1-\lambda} - \frac{K\lambda^K}{1-\lambda^K} \\
&\quad si\, \lambda \neq 1, \qquad = \frac{1+K}{2} \quad sinon.
\end{aligned}
$$

3.3. Résultats avec transfert

Pour étudier le système avec transfert, il est utile de rappeler que la taille de l'espace des états est $K(2^n - 1) + 1$, et il est donc opportun d'utiliser les procédés d'agrégation (*cf.* [12]). L'observation clé est de constater que le nombre total L_t de tâches présentes à l'instant t dans le système évolue comme un processus de naissance et de mort sur $\{0, \ldots, Kn\}$ avec des taux de naissance (de j à $j+1$), notés $\lambda(j)$, et des taux de mort (de j à $j-1$), notés $\mu(j)$. Ces taux sont explicités dans le tableau ci-dessous et comparés avec ceux obtenus pour le modèle $M/M/n/Kn$.

Modèle avec transfert	Modèle $M/M/n/Kn$
$\lambda(j) = \begin{cases} n\lambda & \text{pour } j = 0, \ldots, (K-1)n, \\ (Kn-j)\lambda & \text{pour } j = (K-1)n, \ldots, Kn-1. \end{cases}$	$\lambda(j) = n\lambda.$
$\mu(j) = \begin{cases} j & \text{pour } j = 1, \ldots, n \\ n & \text{pour } j = n+1, \ldots, Kn. \end{cases}$	$\mu(j) = \begin{cases} j & \text{pour } j = 1, \ldots, n \\ n & \text{pour } j = n+1, \ldots, Kn. \end{cases}$

Pour vérifier ces résultats (par exemple pour les taux de naissance pour le modèle avec transfert), il suffit de remarquer que tant qu'il y a moins de $n(K-1)$ tâches dans le système, tous les sites ont moins de $K-1$ tâches et donc toute nouvelle tâche générée par le système est acceptée. S'il y a plus de $n(K-1)$ tâches, tous les sites ayant une charge égale à K provoquent des rejets.

Soit $(p_j)_{j=0\ldots Kn}$ la mesure stationnaire du processus de naissance et de mort $\{L_t, t \geq 0\}$. Cette mesure stationnaire suffit à déterminer celles des lois marginales du processus $\{X_t, t \geq 0\}$, grâce à la proposition 4.

PROPOSITION 4 : *Les probabilités $(P_i(\lambda))_{i=0\ldots K}$ sont reliées aux $(p_j)_{j=0\ldots Kn}$ de la façon suivante :*

$$P_0(\lambda) = \sum_{i=0}^{n-1} \frac{n-i}{n} p_i \quad , \quad P_K(\lambda) = \sum_{i=0}^{n-1} \frac{n-i}{n} p_{Kn-i},$$

$$\forall\, 0 < j < K \, , \quad P_j(\lambda) = \sum_{i=0}^{n} \frac{i}{n} p_{(j-1)n+i} + \sum_{i=1}^{n-1} \frac{n-i}{n} p_{jn+i}.$$

Démonstration. Prenons $0 < j < K$ et $i < n$. Ces équations découlent de la constatation suivante. Lorsqu'il y a $(j-1)n+i$ tâches dans le système, i sites ont j tâches, et $n-i$ ont $j-1$ tâches.

Il est donc nécessaire d'exprimer la valeur de chaque p_j en fonction des paramètres λ, n et K du système. Sans perte de généralité et, pour des raisons de symétrie, la valeur de n est supposée paire, égale à $2q$. Ceci est une contrainte extrêmement faible, puisque les architectures parallèles disposent le plus souvent d'un nombre pair de sites.

En respectant la symétrie du système, et par les techniques habituelles, on obtient alors les formules algébriques suivantes.

$\forall 0 \leq j \leq n$,

$$p_j = \frac{n^j}{j!}\lambda^j p_0 = \frac{n^j}{j!}\frac{1}{\lambda^{Kq-j}}\lambda^{Kq}p_0,$$
$$p_{Kn-j} = \frac{n^j}{j!}\left(\frac{1}{\lambda}\right)^j p_{Kn} = \frac{n^j}{j!}\lambda^{Kq-j}\frac{1}{\lambda^{Kq}}p_{Kn},$$

$\forall n \leq j \leq Kq$,

$$p_j = \frac{n^n}{n!}\lambda^j p_0 = \frac{n^n}{n!}\frac{1}{\lambda^{Kq-j}}\lambda^{Kq}p_0,$$
$$p_{Kn-j} = \frac{n^n}{n!}\left(\frac{1}{\lambda}\right)^j p_{Kn} = \frac{n^n}{n!}\lambda^{Kq-j}\frac{1}{\lambda^{Kq}}p_{Kn}.$$

Posons

$$\rho = p_{Kn}\frac{1}{\lambda^{Kq}} = \lambda^{Kq}p_0.$$

Notons $G(n,\lambda)$ et $F(n,\lambda)$ les sommes

$$G(n,\lambda) = \sum_{j=0}^{n-2}\frac{n^j}{j!}\left(\frac{1}{\lambda}\right)^{Kq-j},$$

$$F(n,\lambda) = \sum_{j=n-1}^{Kq-1}\frac{1}{\lambda^{Kq-j}} = \frac{1}{\lambda^{Kq}}\left(\frac{\lambda^{n-1}-\lambda^{Kq}}{1-\lambda}\right).$$

En utilisant les relations évidentes $\sum_{j=0}^{j=Kn} p_k = 1$ et $\frac{n^n}{n!} = \frac{n^{n-1}}{(n-1)!}$, on obtient,

$$\rho\left[G(n,\lambda)+G\left(n,\frac{1}{\lambda}\right)+\frac{n^n}{n!}\left(1+F(n,\lambda)+F\left(n,\frac{1}{\lambda}\right)\right)\right] = 1.$$

D'où ρ, puis p_0 puis la mesure stationnaire, puis les probabilités $P_j(\lambda)$ d'après la proposition 4, puis la valeur des indices de performance d'après la proposition 2.

4. COMPARAISONS POUR n FIXÉ

Dans cette section nous allons donner, pour fixer les idées, les résultats obtenus pour des valeurs particulières du nombre de sites et de la capacité mémoire K, par exemple $n = 8$, $n = 32$ ou $n = 128$ et $K = 2,\ 3$ ou 6. L'observation du comportement du sytème pour ces valeurs particulières permet d'extrapoler celui obtenu pour les autres valeurs. Les étude numériques ont été effectuées avec *Mathematica*.

4.1. Comparaisons des probabilités stationnaires

Pour des valeurs de n et K fixées, les probabilités stationnaires $P_i(\lambda)$ évoluent en fonction de la valeur du taux d'arrivée λ, et les valeurs de ces probabilités stationnaires sont très différentes selon qu'une politique de transfert est mise ou non en place. Les figures 1 et 2 montrent par exemple les évolutions relatives des probabilités P_0, P_1, P_2, P_3 avec transfert pour $n = 8$, $n = 32$ et $n = 128$.

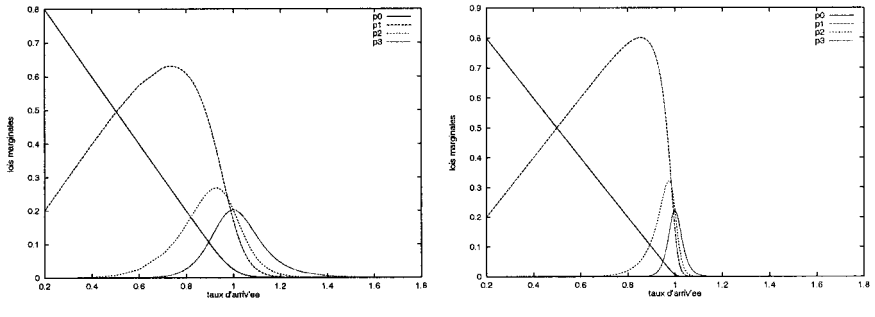

Figure 1. – Évolutions de P_0, P_1, P_2, P_3 fonction de λ pour $n = 8$ puis $n = 32$ sites de capacité $K = 6$.

La valeur de ces probabilités change brutalement autour de $\lambda = 1$ et ce changement est d'autant plus accentué que le nombre n de sites est grand. Dans le cas particulier de $\lambda = 0.8$ et $n = 32$, ces résultats montrent que 98 % des sites ont une charge égale à 0 ou à 1. Autrement dit, le transfert tend à homogénéiser la charge des sites et une proportion de l'ordre de λ d'entre eux ont une charge égale à 1.

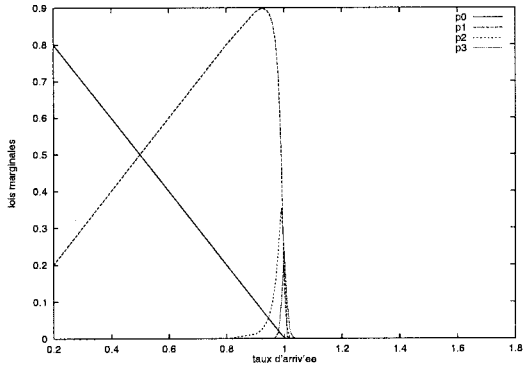

Figure 2. – Évolutions de P_0, P_1, P_2, P_3 fonction de λ pour $n = 128$ sites de capacité $K = 6$.

La différence des comportements du système obtenus avec et sans politique de transfert semble donc assez nette et il semble intéressant d'en mesurer l'incidence sur les indices de performance. Cette étude sera limitée à celle de la probabilité de saturation mémoire et du temps de réponse moyen, l'étude sur les autres indices pouvant s'en déduire grâce à la proposition 2.

4.2. Comparaisons des probabilités de saturation mémoire

Pour des valeurs de n et K fixées, la probabilité de saturation mémoire P_{sat} évolue en fonction de la valeur du taux d'arrivée λ, et de même que précédemment, la valeur de cette probabilité est très différente selon que la politique de transfert est mise ou non en place. La valeur de la capacité K n'altère pas les différences de comportement obtenues. La figure 3 montre

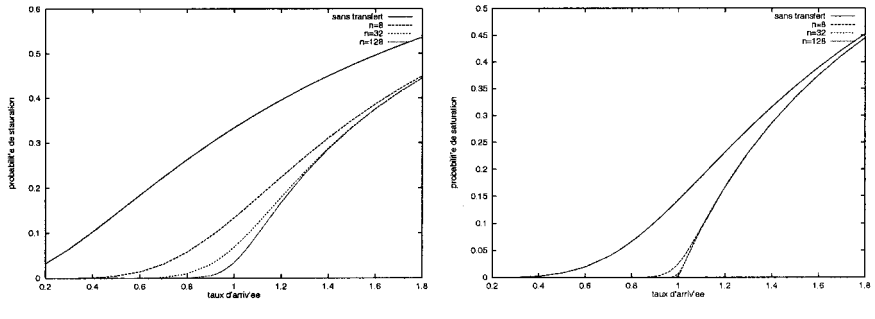

Figure 3. – Évolutions de la probabilité de saturation mémoire pour n sites, $K = 2$ puis $K = 6$.

les évolutions des probabilités de saturation obtenues avec et sans transfert pour $n = 8$, $n = 32$ et $n = 128$ pour des sites de capacité mémoire $K = 2$, et les mêmes évolutions obtenues pour une capacité $K = 6$.

Le transfert améliore significativement cet indice, et la différence $P^*_{\text{sat}} - P_{\text{sat}}$ est d'autant plus petite que le nombre n de sites augmente. Nous reviendrons sur l'étude de cette différence lors de l'étude du comportement asymptotique en section 5.

4.3. Comparaisons des temps de réponse moyens

Rappelons que sans transfert, les sites sont indépendants et le temps de réponse moyen d'une tâche ne dépend pas du nombre n de sites. Avec transfert en revanche le temps de réponse moyen d'une tâche est fortement dépendant du nombre total de sites *i.e.* de n. De plus, pour un nombre n donné de sites, le comportement du temps de réponse moyen dépend de la capacité mémoire K de chaque site. Il convient en effet de distinguer les valeurs $K = 2$ et $K \geq 3$.

La figure 4 montre les évolutions des temps de réponse moyens avec et sans politique de transfert pour n sites de capacité $K = 2$, et les mêmes évolutions obtenues pour une capacité $K = 3$. La figure 5 montre celles obtenues pour une capacité $K = 6$.

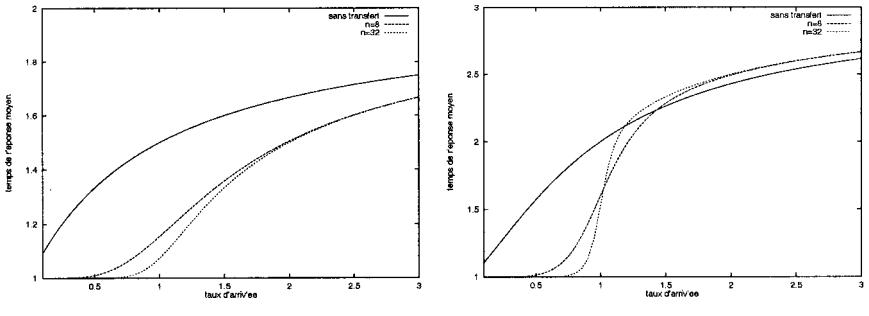

Figure 4. – Évolutions du temps de réponse moyen fonction de λ pour n sites de capacité $K = 2$, $K = 3$.

Ces graphiques appellent quelques commentaires. En effet, il apparaît que le temps de réponse moyen d'une tâche n'est donc pas nécessairement amélioré par le transfert, et il convient ici d'expliciter les comportements observés selon les valeurs des paramètres λ et K.

Pour $\lambda < 1$. La politique de transfert diminue le temps de réponse moyen d'une tâche par rapport à celui obtenu dans la situation sans transfert et

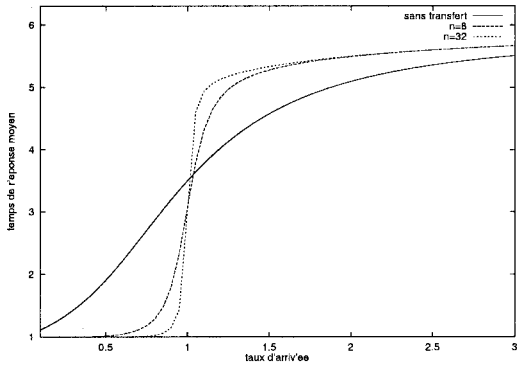

Figure 5. – Évolutions du temps de réponse moyen $K = 6$.

ce, pour toutes les valeurs de la capacité K. L'explication intuitive de ce comportement est simple. En effet, pour un taux d'arrivée des tâches inférieur (strictement) au taux de service, le système est non saturé et d'après les études faites sur les probabilités marginales, un site a très probablement une charge égale soit à 0 soit à 1. Autrement dit, une tâche acceptée dans le système sera vraisemblablement transférée sur un site libre, et sera donc traitée plus rapidement par le système doté d'une politique de transfert que par celui sans politique de transfert.

Pour $\lambda = 1$. Pour toutes les valeurs de la capacité K et du nombre n de sites, la politique de transfert diminue le temps de réponse moyen d'une tâche. Le tableau 1 donne les valeurs de différents temps de réponse obtenus avec et sans transfert pour les différentes valeurs pertinentes des paramètres K et n.

TABLEAU 1
Table de valeurs du temps de réponse moyen pour $\lambda = 1$.

$\lambda = 1$	$K = 2$	$K = 3$	$K = 6$
$\overline{R}^*$	1.5	2.0	3.5
$n = 8$, $\overline{R}$	1.154	1.604	3.078
$n = 32$, $\overline{R}$	1.074	1.533	3.021

Pour $\lambda > 1$. Il convient dans ce cas de distinguer les différents comportements du système selon la valeur de la capacité mémoire K.

Pour $K = 2$. Bien que le système soit saturé, la politique de transfert diminue le temps de réponse moyen d'une tâche. Intuitivement, cela s'explique par le fait que le transfert d'une tâche ne peut être effectué que pour une tâche générée par un site de charge 1 qui la transfère sur un site libre, et la politique de transfert ne modifie le comportement du système que dans cette situation. La valeur moyenne du temps de réponse d'une tâche est donc plus faible avec la politique de transfert que celle obtenue sans transfert. Notons néanmoins que lorsque la valeur du taux d'arrivée λ augmente, cette situation devient de moins en moins probable car la proportion de sites oisifs devient voisine de 0.

Pour $K > 2$. Le système est toujours saturé, et la politique de transfert dans ce cas augmente le temps de réponse moyen d'une tâche. Autrement dit, le temps de réponse moyen d'une tâche générée et acceptée par le système est pénalisé par la politique de transfert. Pour lever cet apparent paradoxe, il faut noter que dans la situation sans transfert, une tâche acceptée par le système a une probabilité non nulle d'être dans les premières à être traitée. La politique de transfert permet en fait d'augmenter le nombre total de tâches acceptées par le système, mais chaque tâche acceptée ne sera pas transférée sur un site sous-chargé et attendra que toutes les tâches qui la précèdent soient traitées. La politique de transfert permet donc de privilégier et d'augmenter l'indice $\overline{T}$, mais le temps de réponse d'une tâche donnée est, en moyenne, pénalisé.

5. COMPORTEMENT ASYMPTOTIQUE DES SYSTÈMES MASSIVEMENT PARALLÈLES

Les études précédentes laissent supposer que le système totalement connecté doté de cette politique de transfert admet un comportement limite lorsque le nombre de sites devient grand. Les formules obtenues analytiquement permettent en effet d'obtenir le comportement asymptotique de ces systèmes dits massivement parallèles. Le théorème de convergence donnant les valeurs limites des lois marginales et des indices de performance obtenues analytiquement pour n tendant vers l'infini est présenté sous forme de tableau dans la section 5.1. La section 5.2 compare ces résultats asymptotiques avec les résultats numériques obtenus pour des valeurs classiques de n (par exemple $n = 32, 64, 128$).

5.1. Théorème de convergence

Pour un taux d'arrivée λ donné, et une capacité mémoire K fixée pour chaque site, les probabilités stationnaires $P_i(\lambda)$ tendent vers une limite quand n tend vers l'infini. Les indices de performance P_{sat} et $\overline{R}$ étant reliés aux probabilités stationnaires par les relations de la proposition 2 admettent aussi des valeurs limites. Toutes ces limites sont données dans le théorème 1.

THÉORÈME 1 : *Avec la politique de transfert, les lois marginales $P_i(\lambda)$ et les indices de performance P_{sat} et $\overline{R}$ admettent les limites suivantes lorsque le nombre de sites devient infini.*

$\lambda < 1$	$\lambda = 1$	$\lambda > 1$
$P_0(\lambda) \to 1-\lambda$	$P_0(\lambda) \to 0$	$P_0(\lambda) \to 0$
$P_1(\lambda) \to \lambda$	$P_1(\lambda) \to 1/(2(K-2))$ $si \quad K \neq 2$	$P_1(\lambda) \to 0$
	$P_1(\lambda) \to 1$ $si \quad K = 2$	$\vdots$
$\vdots$	$\vdots$	$P_i(\lambda) \to 0$
$P_i(\lambda) \to 0$	$P_i(\lambda) \to 1/(K-2)$	$\vdots$
$\vdots$	$\vdots$	$P_{K-1}(\lambda) \to 1/\lambda$
$P_{K-1}(\lambda) \to 0$	$P_{K-1}(\lambda) \to 1/(2(K-2))$	$P_K(\lambda) \to 1-1/\lambda$
$P_K(\lambda) \to 0$	$P_K(\lambda) \to 0$	
$P_{\text{sat}} \to P_{\text{sat}}^{\lim} = 0$	$P_{\text{sat}} \to P_{\text{sat}}^{\lim} = 0$	$P_{\text{sat}} \to P_{\text{sat}}^{\lim} = 1-1/\lambda$
$\overline{R} \to \overline{R}^{\lim} = 1$	$\overline{R} \to \overline{R}^{\lim} = \frac{K}{2}$	$\overline{R} \to \overline{R}^{\lim} = K-1/\lambda$

Commentaires : pour $K > 2$, ces limites présentent une discontinuité en $\lambda = 1$. Un tel comportement irrégulier a déjà été observé dans un modèle différent étudié dans [9], et ces résultats sont cohérents avec ceux qu'ils obtiennent.

Ce comportement asymptotique révèle donc un basculement de comportement intéressant. Lorsque le taux de service est supérieur au taux d'arrivée, une tâche arrivant dans le système est presque toujours acceptée. Elle est presque toujours générée ou transférée sur un site sous-chargé, et son temps de réponse moyen est donc égal à son temps de service moyen, ici égal à 1.

Lorsque le taux de service est égal au taux d'arrivée, une tâche arrivant dans le système est également presque toujours acceptée, mais elle va, en moyenne avoir un temps de réponse égal à $\frac{K}{2}$.

Lorsque le taux de service est inférieur au taux d'arrivée, une tâche peut être refusée, et quand elle est acceptée, elle va presque toujours avoir à attendre beaucoup de temps avant d'être complètement traitée. Plus le taux d'arrivée λ est grand, plus le temps de réponse moyen d'une tâche est proche du temps maximum K et, dans ce cas, la politique de transfert n'apporte aucune amélioration par rapport à celle sans transfert.

Démonstration. Voici une démonstration du théorème 1. Compte tenu de la proposition 1, il suffit de démontrer ce théorème pour $\lambda \leq 1$. Rappelons que le nombre n de sites est supposé pair et que $n = 2q$.

- Pour $P_0(\lambda)$, on obtient,

$$P_0(\lambda) = \sum_{i=0}^{n-1} \frac{n-i}{n} p_i,$$
$$= \frac{(1-\lambda)G(n,\lambda) + \frac{n^n}{n!}(\frac{1}{\lambda^{Kq-n+1}})}{G(n,\lambda) + G(n,\frac{1}{\lambda}) + \frac{n^n}{n!}(1 + F(n,\lambda) + F(n,\frac{1}{\lambda}))}.$$

Il s'agit donc de déterminer le comportement asymptotique de $G(n,\lambda)$. Écrivons

$$\lambda^{Kq} e^{-n\lambda} G(n,\lambda) = \sum_{j=0}^{j=n-2} \frac{(n\lambda)^j}{j!} e^{-n\lambda}.$$

Ceci est la valeur au point $n-2$ de la fonction de répartition d'une loi de Poisson de paramètre λn. D'après le théorème central limite, il vient

$$\lim_{n\to+\infty} \lambda^{Kq} e^{-n\lambda} G(n,\lambda) = 1 \quad si \quad \lambda < 1$$
$$= \frac{1}{2} \quad si \quad \lambda = 1$$
$$= 0 \quad si \quad \lambda > 1.$$

Il convient donc de distinguer les 2 situations possibles $\lambda = 1$ et $\lambda < 1$.

Pour $\lambda = 1$, on obtient (formule de Stirling),

$$\frac{n^n}{n!} \times \frac{1}{G(n,1)} \sim \frac{1}{2\sqrt{2\pi n}}.$$

Pour $\lambda < 1$, on obtient,

$$\frac{G(n,1)}{G(n,\lambda)} \sim \frac{1}{2}\left(\frac{\lambda^{\frac{K}{2}}}{e^{\lambda-1}}\right)^n.$$

D'où

$$\lim_{n\to+\infty} \frac{G(n,1)}{G(n,\lambda)} = 0,$$

et par suite

$$\lim_{n\to+\infty} \frac{\frac{n^n}{n!}\lambda^{Kq-n+1}}{G(n,\lambda)} = 0.$$

Or, pour $\lambda < 1$,

$$G(n,\frac{1}{\lambda}) \leq G(n,1).$$

Donc

$$\lim_{n\to+\infty} \frac{G(n,1/\lambda)}{G(n,\lambda)} = 0.$$

Ces résultats permettent de conclure :

Pour $\lambda = 1$: le terme dominant de $P_0(1)$ est $G(n,1)$ au dénominateur, et par suite

$$\lim_{n\to+\infty} P_0(1) = 0.$$

Pour $\lambda < 1$: le terme dominant du numérateur et du dénominateur de $P_0(\lambda)$ est $G(n,\lambda)$. D'où

$$\lim_{n\to+\infty} P_0(\lambda) = 1 - \lambda.$$

- Pour $P_1(\lambda)$, on obtient,

$$\begin{aligned} P_1(\lambda) &= \sum_{i=0}^{n} \frac{i}{n} p_i + \sum_{i=1}^{n-1} \frac{n-i}{n} p_{n+i}, \\ &= \frac{\lambda G(n,\lambda) + \frac{n^n}{n!}(\frac{1}{\lambda^{Kq-n+1}})}{G(n,\lambda) + G(n,\frac{1}{\lambda}) + \frac{n^n}{n!}(1 + F(n,\lambda) + F(n,\frac{1}{\lambda}))} \\ &\quad + \sum_{i=1}^{n-1} \frac{n-i}{n} \frac{n^n}{n!}\left(\frac{1}{\lambda^{Kq-(n+i)}}\right)\rho. \end{aligned}$$

En utilisant les mêmes arguments, on obtient

Pour $\lambda < 1$: le terme dominant des numérateurs et du dénominateur de $P_1(\lambda)$ est $G(n, \lambda)$. D'où

$$\lim_{n \to +\infty} P_1(\lambda) = \lambda.$$

Pour $\lambda = 1$: après calculs, pour $K \neq 2$, on obtient

$$P_1(1) = \frac{G(n,1) + \frac{n^n}{n!}(\frac{n}{2} + \frac{1}{2})}{2G(n,1) + \frac{n^n}{n!}[(K-2)n + 2]}.$$

Or,

$$\lim_{n \to +\infty} \frac{G(n,1)}{n\frac{n^n}{n!}} = 0,$$

Et par suite, pour $K \neq 2$,

$$\lim_{n \to +\infty} P_1(1) = \frac{1}{2(K-2)}.$$

- Autres résultats : les autres résultats obtenus pour $\lambda < 1$ découlent de ceux obtenus pour $P_0(\lambda)$ et $P_1(\lambda)$. Enfin, pour $\lambda = 1$, la symétrie du graphe de transition du processus de naissance et de mort du processus agrégé $\{L_t, t \geq 0\}$ et les relations algébriques de la proposition 2 permettent d'écrire les équations suivantes,

$$\begin{aligned} &P_0(1) = P_K(1), \\ &P_1(1) = P_{K-1}(1), \\ &P_2(1) = P_3(1) = \ldots = P_{(K-2)n}(1). \end{aligned}$$

D'où les autres résultats obtenus pour $\lambda = 1$ avec $K > 2$ ou $K = 2$. ■

Remarque : une justification plus intuitive de ce théorème peut être donnée de la façon suivante.

Le processus de naissance et de mort $\{L_t, t \geq 0\}$ se comporte sur l'intervalle $\{0 \ldots, n\}$ comme une file $M/M/\infty$, avec un taux d'arrivée $n\lambda$ et un taux de service 1. La mesure stationnaire de cette file suit une loi de Poisson de paramètre $n\lambda$. Pour $\lambda < 1$, la probabilité que $\{L_t, t \geq 0\}$ dépasse n tend vers 0 quand n augmente. Ainsi la mesure stationnaire du processus $\{L_t, t \geq 0\}$ tend vers une distribution de Poisson dont l'espérance est $n\lambda$. Pour des systèmes massivement parallèles (n grand), la plupart des sites ont une charge égale à 0 ou à 1. En moyenne, une proportion λ d'entre

eux ont une charge égale à 1. D'où les résultats dans le cas des systèmes non saturés ($\lambda < 1$). Dans le cas $\lambda = 1$, le même argument concernant la distribution de Poisson permet de déduire que la probabilité qu'un site soit libre tend vers 0. Les mêmes arguments que ceux utilisés dans la démonstration permettent alors de conclure.

5.2. Comparaisons entre les résultats numériques et les résultats asymptotiques

Les résultats numériques de cette section ont été obtenus avec *mathematica*. Le but de ces calculs est de montrer que les résultats théoriques obtenus dans le théorème 5.1 donnent une bonne approximation du comportement du système pour les nombres classiques de sites mis en réseau dans les machines parallèles, à savoir $n = 32, 64, 128, 256$ ou plus.

5.2.1. *Probabilité de saturation*

Pour toutes les valeurs du taux d'arrivée λ données, la probabilité de saturation mémoire est une fonction décroissante du nombre n de sites. Autrement dit, avec la politique de transfert, ajouter un site dans un réseau totalement connecté diminue toujours la probabilité de saturation du système et tend à la stabiliser vers sa valeur limite $P_{\text{sat}}^{\lim}$. Les résultats numériques montrent que la convergence vers cette valeur limite est très rapide, et que la convergence la plus lente est obtenue pour $\lambda = 1$. La figure 6 montre l'évolution de cette probabilité de saturation en fonction du nombre n de sites de capacité $K = 6$.

Pour plus de précision, le tableau 2 donne par exemple les valeurs $P_{\text{sat}} - P_{\text{sat}}^{\lim}$ dans cette situation.

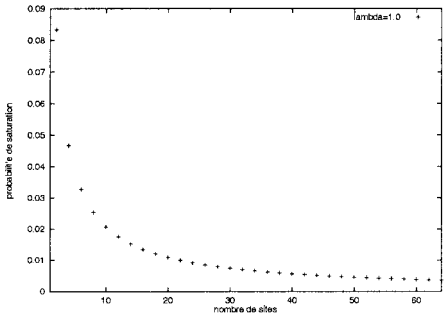

Figure 6. – Probabilité de saturation pour $\lambda = 1$ fonction du nombre de sites (capacité $K = 6$).

TABLEAU 2
Table de la différence $P_{\rm sat} - P_{\rm sat}^{\rm lim}$ pour n sites de capacité $K = 6$.

λ	1
$n = 16$	0.0134
$n = 32$	0.0070
$n = 64$	0.0036
$n = 128$	0.0018

Autrement dit, quand on dispose de 32 sites, ajouter un site augmente encore la probabilité qu'une tâche soit acceptée dans le système, mais de façon moins significative, puisque le gain obtenu est inférieur à 0.007.

5.2.2. *Temps de réponse moyen*

Le temps de réponse moyen pour un taux d'arrivée λ et une capacité K donnés est une fonction monotone de n, dont le sens dépend des valeurs de λ et de K.

Pour $\lambda < 1$, augmenter le nombre de sites diminue le temps de réponse moyen et ce temps moyen tend rapidement vers 1 ce qui correspond au temps de service moyen.

Pour $\lambda = 1$, le temps de réponse moyen diminue quand le nombre n de sites augmente, et tend vers $\frac{K}{2}$. La figure 7 montre, par exemple, les courbes obtenues pour des taux d'arrivée $\lambda = 0.8$ et $\lambda = 1$ avec $n = 2q$ sites de capacité $K = 6$.

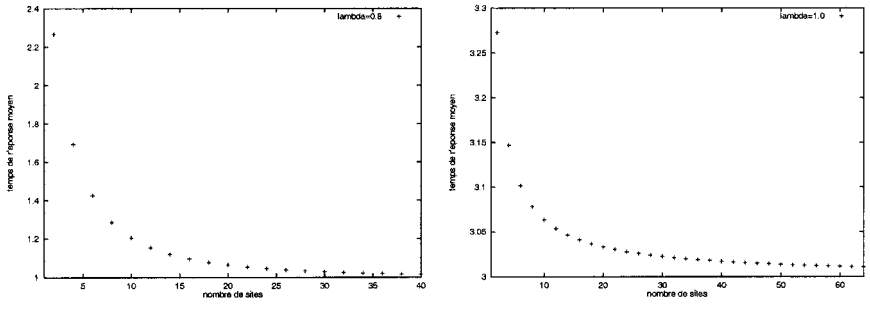

Figure 7. – Temps de réponse moyen fonction du nombre de sites (capacité $K = 6$) pour $\lambda = 0.8$ puis $\lambda = 1$.

Pour $\lambda > 1$, comme il a déjà été mentionné au 4.3, l'évolution du temps de réponse moyen en fonction de n dépend de la capacité mémoire K.

Le temps de réponse moyen est une fonction monotone de n qui est décroissante si la capacité mémoire est $K = 2$ et est croissante dans tous les autres cas. La figure 8 donnent par exemple les résultats obtenus pour un taux d'arrivée $\lambda = 1.25$ avec les capacités $K = 2$, $K = 3$ et $K = 6$ en fonction du nombre n de sites.

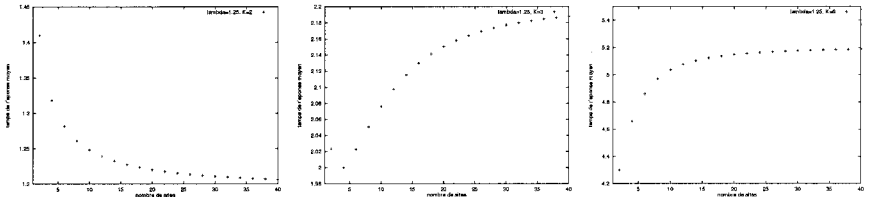

Figure 8. – Temps de réponse moyen fonction du nombre de sites, $\lambda = 1.25$, $K = 2$, $K = 3$ et $K = 6$.

Dans tous les cas, la stabilité autour de la limite est aussi rapidement atteinte. Le tableau 3 donne les valeurs $|\overline{R} - \overline{R}^{\lim}|$ pour un nombre n de sites de capacité $K = 6$ et un taux d'arrivée λ donné, avec $n = 16, 32, 64, 128$.

TABLEAU 3
Table de la différence $|\overline{R} - \overline{R}^{\lim}|$ *(capacité* $K = 6$*).*

λ	0.6	0.8	1	1.2	1.4
$n = 16$	0.0065	0.0953	0.0409	0.121	0.0232
$n = 32$	0.004	0.0251	0.0212	0.0369	0.0038
$n = 64$	$< 10^{-4}$	0.0044	0.0109	0.0084	0.0003
$n = 128$	$< 10^{-4}$	0.0004	0.0056	0.0011	$< 10^{-4}$

De même que pour la probabilité de saturation mémoire, la valeur $\lambda = 1$ est celle pour laquelle le système converge le plus lentement, mais même pour cette valeur et pour un système ayant au moins 32 sites, le temps de réponse moyen d'une tâche peut être estimé, avec l'approximation obtenue dans le tableau 3, par sa valeur asymptotique $\overline{R}^{\lim}$.

6. CALCULS D'OPTIMISATION

6.1. Bornes supérieures sur le bénéfice obtenu avec le transfert

Intuitivement, la politique de transfert étudiée correspond à une situation idéale et les gains obtenus dans cette situation peuvent être considérés comme des bornes supérieures sur le bénéfice que l'on peut attendre d'un

réel transfert de charge. D'autres bornes intéressantes sont celles obtenues avec la file $M/M/n/Kn$ qui modélise aussi un partage idéal. Comme dans la section 4, cette étude sera limitée à celle de P_{sat} et $\overline{R}$.

Probabilité de saturation mémoire. Le gain obtenu sur la probabilité de saturation mémoire grâce à la politique de transfert est $P^*_{\text{sat}} - P_{\text{sat}}$. Pour des sites de capacité K donnée, et pour un taux d'arrivée λ donné, le meilleur gain est obtenu pour n infini, et la valeur de ce gain est

$$\begin{aligned}\text{Gain}^{P_{\text{sat}}}_{\max}(\lambda, K) &= \lambda^K \frac{1-\lambda}{1-\lambda^{K+1}} \quad \text{si } \lambda \leq 1,\\ &= \frac{1}{K+1} \quad \text{si } \lambda = 1,\\ &= \lambda^K \frac{1-\lambda}{1-\lambda^{K+1}} + \frac{1}{\lambda} - 1 \quad \text{si } \lambda > 1.\end{aligned}$$

Pour une capacité K donnée, le meilleur gain est donc obtenu pour $\lambda = 1$ et vaut

$$\text{Gain}^{P_{\text{sat}}}_{\max}(K) = \frac{1}{K+1}.$$

C'est donc lorsque le taux de service est égal au taux d'arrivée que la politique de transfert permet en moyenne de mieux gérer l'ensemble des mémoires du système. Le gain obtenu sur la probabilité de saturation mémoire est d'autant plus important que la capacité K des sites est petite. Ce gain est donc au mieux égal à $1/3$, lorsque la capacité mémoire est égale à 2.

Temps de réponse moyen. Comme nous l'avons déjà constaté, le gain obtenu pour le temps de réponse moyen grâce à la politique de transfert dépend de λ de n, et de K, et il convient de distinguer les résultats selon les valeurs de ces paramètres.

Pour $\lambda < 1$ et une capacité K donnés, le gain obtenu est $\overline{R}^* - \overline{R}$, et le meilleur gain, obtenu pour n infini, est

$$\text{Gain}^{\overline{R}}_{\max}(\lambda, K) = \frac{\lambda}{1-\lambda} - \frac{K\lambda^K}{1-\lambda^K}.$$

Le meilleur gain est donc obtenu pour $\lambda \to 1$ et vaut

$$\text{Gain}^{\overline{R}}_{\max}(K) = \frac{K-1}{2}.$$

Autrement dit, plus la capacité K est grande, plus le gain sur le temps de réponse moyen d'une tâche est important. Ce résultat est intuitif car pour λ très proche de 1 mais en régime non saturé ($\lambda < 1$), une tâche générée dans le système sans transfert de capacité K va attendre beaucoup, tandis qu'elle va en moyenne être traitée la première quand la politique de transfert est mise en œuvre.

Pour $\lambda = 1$, le gain obtenu pour une capacité K est toujours $\overline{R}^* - \overline{R}$, et le meilleur gain obtenu pour n infini est

$$\mathrm{Gain}^{\overline{R}}_{\max}(1, K) = \frac{1}{2}.$$

Notons que cette limite est indépendante de la capacité mémoire K. Dans la situation sans transfert, la file $M/M/1/K$ est saturée. Le transfert améliore donc le temps de réponse moyen d'une tâche, mais ne permet pas de gagner plus de $50\ \%$ du temps de service moyen, quelle que soit la capacité K des sites du système.

Pour $\lambda > 1$, le temps de réponse moyen d'une tâche générée par le système avec une politique de transfert peut être supérieur à celui d'une tâche générée par le système sans transfert. Pour une capacité K donnée et un nombre infini de sites, la différence $\overline{R} - \overline{R}^*$ est

$$K - \frac{1}{\lambda} - \frac{\lambda}{1-\lambda} + \frac{K\lambda^K}{1-\lambda^K} - 1,$$

et cette différence peut être positive ou négative selon les valeurs de la capacité mémoire K de chaque site. Il apparaît que pour $K = 2$, le transfert diminue toujours le temps de réponse moyen tandis que pour $K > 2$, le temps de réponse moyen avec une politique de transfert devient supérieur au temps de réponse moyen sans transfert dès que le taux d'arrivée devient supérieur au taux de service (ici $\lambda = 1$). Dans cette situation, la différence entre les temps de réponse moyen obtenus sans transfert et avec une politique de transfert accuse donc un brusque changement de comportement en $\lambda = 1$, et ce changement est d'autant plus significatif que la capacité mémoire K est grande.

Le meilleur bénéfice possible sur le temps de réponse moyen d'une tâche grâce à la politique de transfert est donc obtenu pour des valeurs du taux d'arrivée proche du taux de service, en régime non saturé. Dans cette situation, le bénéfice maximum est $\frac{K-1}{2}$,

et celui-ci est donc d'autant plus important que la capacité K est grande, et le temps d'attente d'une tâche est alors considérablement réduit grâce à la politique de transfert.

Pour souligner ces différences de comportement, la figure 9 montre le gain obtenu pour la probabilité de saturation mémoire pour $K = 6$ et les évolutions de la différence $\overline{R}^* - \overline{R}$ en fonction du taux d'arrivée λ, pour un nombre infini de sites de capacité mémoire $K = 2$, $K = 3$ et $K = 6$.

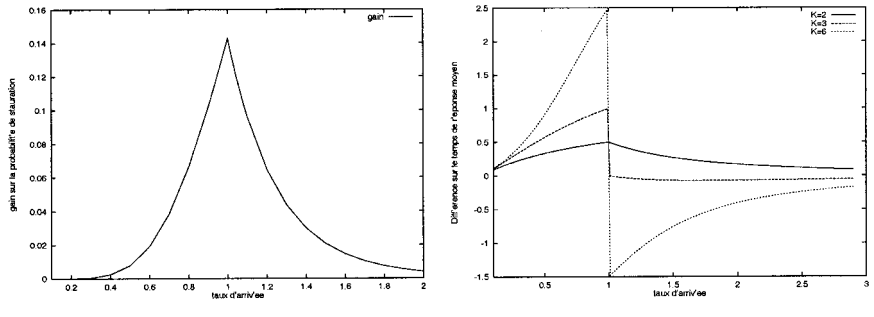

Figure 9. – Gain sur P_{sat} pour $K = 6$ puis différence $\overline{R}^* - \overline{R}$ pour $K = 2$, $K = 3$ et $K = 6$.

6.2. Optimisation du taux d'arrivée

Une application possible de cette modélisation est par exemple, de comparer les valeurs minimales que doit avoir le taux d'arrivée λ pour obtenir un débit global fixé pour chaque site.

Ce débit global (throughput dans la littérature anglaise) pour chaque site est le nombre moyen de tâches traitées par unité de temps (*cf.* [3]). Avec ou sans politique de transfert, lorsque le temps de service moyen est pris comme unité de temps, ce débit global, noté ici δ est : $\delta = 1 - P_0(\lambda)$. Afin de garantir une efficacité maximum ou un rendement donné important de chaque site, il faut assurer un taux d'arrivée des tâches suffisamment important, et ces valeurs diffèrent selon que la politique de transfert est mise en place ou non.

Pour 32 sites de capacité $K = 6$, et pour quelques débits particuliers, le tableau 4 donne les valeurs minimales que doit avoir le taux d'arrivée λ pour garantir ce débit dans les situations avec et sans transfert.

Afin d'assurer un débit moyen constant donné pour chaque site, proche de l'efficacité maximum de ce site ($\mu = 1$), il est donc nécessaire, d'assurer un taux d'arrivée significativement élevé (par rapport au taux de service) sans transfert, tandis qu'avec transfert, un taux d'arrivée égal au débit moyen

TABLEAU 4
Valeurs minimales du taux d'arrivée λ pour un débit δ souhaité avec 32 sites de capacité $K = 6$.

débit δ souhaité	λ minimale sans transfert	λ minimale avec transfert
0.95	1.34201	0.95
0.90	1.11712	0.90
0.85	0.98369	0.85

souhaité est suffisant. Sans politique de transfert, il y a donc beaucoup de pertes de tâches par rapport au nombre de tâches générées, tandis qu'avec la politique de transfert, la perte de tâches est infime, même pour des valeurs du taux d'arrivée proches du taux de service moyen.

6.3. Dimensionnement de la capacité mémoire

Une autre problématique intéressante est de trouver une valeur de la capacité mémoire permettant de garantir une probabilité de saturation mémoire faible pour les valeurs du taux d'arrivée inférieures au taux de service (ici pour $\lambda \leq 1$). Pour un nombre n de sites et une valeur du taux d'arrivée λ donnés, la probabilité de saturation mémoire est une fonction décroissante de la capacité mémoire K. Pour obtenir une probabilité de saturation mémoire inférieure à un seuil donné la capacité mémoire requise est d'autant plus grande que le taux d'arrivée est proche du taux de service.

Dans les deux situations, avec et sans politique de transfert, et dans le cas où le taux de service λ est égal au taux de service de moyen, ici $\lambda = 1$, le tableau 5 donne les valeurs de la capacité mémoire nécessaire pour obtenir une probabilité de rejet inférieure à 0.001.

TABLEAU 5
Dimensionnement de la capacité mémoire K pour obtenir $P_{\mathrm{sat}} < 0.001$.

λ	K sans transfert	K avec transfert, $n = 8$	K avec transfert, $n = 32$
1.0	999	130	33

Pour obtenir une probabilité de saturation mémoire raisonnablement proche de zéro, la politique de transfert permet donc de réduire considérablement la taille de la capacité mémoire de chaque site.

7. ESTIMATION DU COÛT DES TRANSFERTS

7.1. Nombre moyen de transferts par unité de temps

Dans la modélisation étudiée ici, le temps des protocoles des communications qui précèdent le transfert et celui du transfert lui-même ont été négligés par rapport au temps de service moyen. Il est toutefois opportun, pour un système doté de cette politique de transfert d'estimer le nombre moyen de transferts effectués par unité de temps. Avec d'autres hypothèses de modélisations, certaines études permettent de tenir explicitement compte de certains délais de communications (*cf.* [10]).

Soit N_t la variable aléatoire égale au nombre de transferts intervenus entre l'instant initial 0 et l'instant t.

Le processus $\{N_t, t \geq 0\}$ est un processus de Poisson à environnement Markovien noté dans la littérature MMPP (Markov modulated Poisson process). Quand i sites ont une charge égale à j et les $n-i$ autres sites ont une charge égale à $j-1$ ($1 \leq j \leq K$). Avec un temps d'exécution moyen égal à 1 le taux de passage de n à $n+1$ de ce processus est $\lambda i + (n-i)$. Le théorème exposé dans la section "the counting function" de [6] permet d'obtenir le nombre moyen de transferts par unité de temps en fonction des probabilités stationnaires $(p_j)_{j=0\ldots Kn}$ du processus $\{L_t, t \geq 0\}$. Ce nombre moyen de transferts par unité de temps, $\overline{N}_{\mathrm{tra}}$ s'exprime par la relation suivante (avec les notations de 3.3).

$$\overline{N}_{\mathrm{tra}} = \lim_{t \to +\infty} E\left(\frac{N_t}{t}\right) = \sum_{j=1}^{K} \left(\sum_{i=1}^{n-1} \lambda i p_{(j-1)n+i} \right) + \sum_{j=1}^{K} \left(\sum_{i=1}^{n-1} (n-i) p_{(j-1)n+i} \right).$$

Les tableaux 6 et 7 donnent les valeurs obtenues pour ce nombre de transferts moyen pour des sites de capacité mémoire $K = 2$ et $K = 6$ avec des valeurs variables du taux d'arrivée λ et du nombre n de sites.

TABLEAU 6
Nombre moyen de transferts par unité de temps pour une capacité mémoire $K = 2$.

λ	0.7	0.8	0.9	1	1.1	1.2	1.3
$n = 32$	25.23	26.63	28.24	29.80	31.22	32.59	34.01
$n = 64$	50.55	53.65	57.41	60.86	63.53	65.85	68.36
$n = 128$	101.12	107.50	115.87	123.54	128.32	132.17	136.85

TABLEAU 7
Nombre moyen de transferts par unité de temps pour une capacité mémoire $K = 6$.

λ	0.7	0.8	0.9	1	1.1	1.2	1.3
$n = 32$	25.25	26.75	28.71	30.88	31.79	32.83	34.11
$n = 64$	50.56	53.72	57.96	62.84	64.23	66.03	68.40
$n = 128$	101.12	107.51	116.33	126.82	128.96	132.24	136.85

Ce nombre moyen de transferts par unité de temps dépend assez peu de la capacité mémoire. A condition que le temps de transfert d'une tâche soit petit devant sa durée moyenne d'exécution, on remarque aussi que ce nombre moyen de transferts par unité de temps est de l'ordre de la taille n du système. Par suite, la surcharge du réseau de communication due à la politique de transfert reste acceptable. Les contentions dans les réseaux n'influeront que très peu sur les performances globales du système.

7.2. Nombre moyen de transferts pour une tâche donnée

Avec cette politique de modélisation de la politique de transfert, une tâche générée et acceptée par le système, peut être transférée plusieurs fois d'un site à un autre. Il est donc utile d'estimer le nombre moyen de transferts que cette tâche va subir pendant son séjour dans le réseau. Ce nombre, que nous noterons $\overline{N}_{\mathrm{tts}}$, peut s'obtenir à partir du nombre moyen de transferts $\overline{N}_{\mathrm{tra}}$ et du temps moyen de réponse d'une tâche $\overline{R}$, et du débit moyen du système $\overline{T}$. Le nombre moyen de transferts $\overline{N}_{\mathrm{tts}}$ que subit une tâche pendant son séjour dans le système est en effet :

$$\overline{N}_{\mathrm{tts}} = \frac{\overline{N}_{\mathrm{tra}}}{\overline{T}}(\overline{R} - 1).$$

Les tableaux 8 et 9 donnent les valeurs obtenues pour ce nombre moyen de transferts que subit une tâche pendant son séjour pour des capacités mémoire $K = 2$ et $K = 6$ avec des valeurs variables du taux d'arrivée λ et du nombre n de sites.

TABLEAU 8
Nombre moyen de transferts d'une tâche donnée pour n sites de capacité mémoire $K = 2$.

λ	0.7	0.8	0.9	1	1.1	1.2	1.3
$n = 32$	0.00	0.01	0.04	0.07	0.13	0.19	0.25
$n = 64$	0.00	0.00	0.02	0.05	0.11	0.18	0.25
$n = 128$	0.00	0.00	0.01	0.04	0.10	0.17	0.25

Avec une capacité mémoire égale à 2, rappelons qu'une tâche arrivant dans le système est transférée si et seulement si elle arrive sur un site de charge 1 tandis qu'il existe un site libre. Dans une telle situation, une charge subit donc au plus un transfert, et le nombre de transferts que subit une tâche semble donc acceptable. Notons la décroissance de ce nombre particulièrement quand $\lambda > 1$, mais rappelons que dans cette situation, même pour de petites valeurs de n, la probabilité qu'un site soit oisif est voisine de zéro, et la situation de transfert d'une tâche est d'autant moins probable que n est grand.

TABLEAU 9
Nombre moyen de transferts d'une tâche donnée pour n sites de capacité mémoire $K = 6$.

λ	0.7	0.8	0.9	1	1.1	1.2	1.3
$n = 32$	0.00	0.03	0.14	1.96	3.91	4.24	4.50
$n = 64$	0.00	0.00	0.05	1.98	4.05	4.29	4.52
$n = 128$	0.00	0.00	0.01	1.99	4.11	4.30	4.52

Pour une capacité mémoire $K = 6$, une tâche entrant dans le système peut subir entre 0 et 5 transferts avant d'être traitée. Pour des valeurs du taux d'arrivée inférieures au taux de service, elle va subir en moyenne très peu de transferts. Pour un taux d'arrivée égal au taux de service, ici $\lambda = 1$, le nombre moyen de transferts de cette tâche reste « raisonnable » de l'ordre de deux. Par contre, pour des valeurs du taux d'arrivée supérieures au taux de service, ici $\lambda > 1$, ce nombre moyen de transferts d'une tâche est important, de l'ordre de 4, voire 5 transferts, c'est-à-dire proche du maximum de transferts possibles. Notons enfin que pour cette capacité $K = 6$ et $\lambda > 1$, ce nombre augmente quand le nombre de sites augmente car la probabilité de transfert d'une tâche augmente. Le coût de la politique de transfert dans cette situation peut alors s'avérer important.

8. COMPORTEMENT ASYMPTOTIQUE POUR UNE CAPACITÉ MÉMOIRE NON LIMITÉE

Il peut être intéressant d'étudier le comportement du système doté de cette politique de transfert lorsque la capacité mémoire de chaque site est idéalement non limitée. Le régime d'équilibre n'a alors de sens que lorsque le taux d'arrivée est strictement inférieur au taux de service, ici $\lambda < 1$.

Sans politique de transfert, on retrouve alors les résultats classiques de n files $M/M/1$ indépendantes.

Avec une politique de transfert, le système se comporte alors comme une file $M/M/n$ pour laquelle le temps inter-arrivées suit une loi exponentielle de paramètre $n\lambda$. La probabilité de saturation mémoire n'a plus de sens dans ce contexte, et le temps de réponse moyen s'exprime de la façon suivante (avec les notations du 3.3) :

PROPOSITION 5 : *Pour une capacité mémoire $K = \infty$, le temps de réponse moyen en fonction du taux d'arrivée λ et du nombre n de sites est :*

$$\overline{R} = \frac{G(n,\lambda) + \frac{n^n}{n!}\left(\frac{\lambda^{n-1}}{1-\lambda} + \frac{\lambda^n}{n(1-\lambda)^2}\right)}{G(n,\lambda) + \frac{n^n}{n!}\frac{\lambda^{n-1}}{1-\lambda}}.$$

Pour un système massivement parallèle, on obtient :

$$\lim_{n\to+\infty} \overline{R} = 1.$$

Une tâche arrivant dans un tel système va donc presque sûrement être traitée immédiatement. Cela correspond à un comportement limite idéal pour lequel la politique de transfert est éminemment bénéfique.

9. CONCLUSION

Le transfert de charge est une pratique essentielle pour obtenir les meilleures performances possibles d'une machine parallèlle, mais compte tenu du coût en temps et en communication d'un réel transfert, celui-ci doit être effectué à bon escient. En imaginant une situation idéale où le transfert est déclenché dès que la différence de charge entre 2 sites est supérieure à deux, et en admettant que le transfert est immédiat et instantané, nous ne traduisons pas toute la complexité des problèmes posés par le transfert, mais ce modèle d'équilibrage global de la charge permet de mesurer (en terme d'espérance) l'impact du transfert sur les indices de performances utilisés dans le domaine (proportion de sites ayant i tâches, probabilité de saturation mémoire, temps de réponse moyen). On peut ainsi obtenir les bornes supérieures des bénéfices que l'on peut attendre d'un réel transfert, comprendre dans quelles situations le transfert de charge peut être moins intéressant que la politique qui consiste à les interdire et examiner en fonction de la valeur des paramètres du modèle l'opportunité du transfert.

Avec un temps d'exécution moyen pris comme unité de temps, cette étude permet de montrer les résultats suivants. Pour obtenir de réels bénéfices

grâce au transfert il est opportun d'effectuer les transferts pour des valeurs de λ comprises entre 0.70 et 0.95. En effet, en dessous de ces valeurs, le transfert n'améliore pas considérablement les performances du système, et au dessus de ces valeurs, le système est de toute façon surchargé et le coût réel du transfert risque de contre-balancer les éventuels bénéfices apportés par le transfert. Pour s'assurer que l'on est dans les fourchettes propices au transfert, on peut utiliser la proportion de sites libres. En effet, celle-ci reste très élevée tant que $\lambda < 0.95$, puis avoisine zéro quand le taux d 'arrivée λ dépasse cette valeur. Le fait le plus marquant que l'on peut observer grâce aux courbes obtenues dans la section 4 est que le passage de 8 à 32 sites n'améliore pas de manière significative les résultats sur les indices de performances P_{sat} et $\overline{R}$. Ceci laisse à penser qu'en ayant des contraintes de localité faible (par exemple quand chaque site est connecté à 8 autres sites seulement) des performances similaires à celle d'un réseau totalement connecté peuvent être espérées. Avec un graphe idéalement complet, le nombre de sites peut être limité à 32 puisque les meilleures performances sont pratiquement atteintes pour ce nombre de sites et que les gains obtenus grâce au transfert au-delà de 32 sites ne justifient sans doute pas de nouveaux investissements. Une politique de transfert idéale entre 32 sites permet alors de dimensionner la capacité des files d'attente de façon raisonnable afin de minimiser la probabilité de saturation mémoire en régime non saturé.

Ajouter un délai pour tenir compte du temps de décision d'un transfert et un délai pour le transfert a été fait dans le cas de deux sites pour $K = 2$ (*cf.* [2]), mais les calculs semblent prohibitifs dans le cas général. Citons néanmoins les modèles de Malyshev et Robert [9] qui pénalisent le transfert mais avec des capacités égales à 1, et ceux de Mirchananey *et al.* [10] qui pénalisent également le transfert dans des cas particuliers. Dans un cadre général cette prise en compte s'avèrera sans doute délicate à résoudre analytiquement, mais des simulations peuvent donner des informations très pertinentes. On peut aussi envisager l'étude de cet algorithme pour d'autres lois de distribution du temps de service et du processus d'arrivée des tâches sur chaque site. On peut encore faire évoluer ce modèle en imaginant d'autres architectures de sites (par exemple des réseaux toriques à une ou deux dimensions). Les outils théoriques utilisés seront alors issus de la théorie des systèmes de particules.

RÉFÉRENCES

1. A. T. Barucha-Reid, *Elements of the theory of Markov processes and their Applications*, Mc Graw-Hill, Londres, 1960.

2. M. Béguin, *Transfert de charge dans les machines parallèles*, Rapport technique 6, MAI-IMAG, Grenoble, Octobre 1994.
3. D. P. Bertzekas et J. N. Tsitsiklis, *Parallel and distributed computation*, Prentice-Hall, 1989.
4. P. Brémaud, *Introduction aux chaînes de Markov et aux files d'attente*, Polycopié de l'ENSTA, 1982.
5. D. J. Evans et W. U. N. Butt, *Dynamic Load Balancing using Task-transfer Probabilities*, Parallel Computing, 1993, *19*, p. 897–916.
6. W. Fisher et K. Meier-Hellstern, *The Markov-modulated Poisson process (MMPP) cookbook*, Performance Evaluation, 1993, *18*, p. 149–171.
7. L. Kleinrock, *Queueing systems: Theory*, volume 1, J. Wiley & Sons, 1975.
8. J. D. C. Little, *A proof of the queueing formula* $L = \lambda W$, Oper. Res., 1961, *9*, p. 383–387.
9. V. Malyshev et P. Robert, *Phase transition in a loss load sharing model*, Ann. Appl. Probab., 1994, *4*, p. 1161–1176.
10. R. Mirchananey, D. Towsley et J. A. Stankovic, *Analysis of the effects of delays on load sharing*, IEEE Trans. on Computers, 1989, *C-38*, p. 1513–1525.
11. B. Plateau, *Algorithmique parallèle et partage de charge*, Rapport technique, APACHE-IMAG, 1994.
12. M. Rosenblatt, *Functions of a Markov process that are Markovian*, Journal of Mathematics and Mechanics, 1959, *8*, p. 585–596.
13. S. M. Ross, *Introduction to probability models, fourth edition*, Academic Press, INC, Londres, 1989.
14. F. Spies, H. Guyennet et M. Trehel, *Modeling and simulation of dynamic load balancing using queueing theory*, Parallel Algorithms and Applications, Gordon and Breach science publishers, 1995, *5*, p. 199–218.
15. M. S. Squillante et R. D. Nelson, *Analysis of task migration in shared-memory multiprocessors*, In Proceedings of the ACM SYGMETRICS Conference on Meassurement and Modeling of Computer Systems, 1991, p. 143-155.
16. J. Walrand, *Introduction to Queueing Networks*, Prentice-Hall, 1989.