À la racine du parallélisme

Thomas Bonald, Céline Comte, Fabien Mathieu

À la racine du parallélisme

Thomas Bonald[1], Céline Comte[21] et Fabien Mathieu[2] †

[1] *Télécom ParisTech, Université Paris-Saclay, France*
[2] *Nokia Bell Labs, France*

Nous considérons un cluster de serveurs traitant des requêtes en parallèle. Si les clients ont en général intérêt à ce que leurs requêtes soient traitées par le plus grand nombre de serveurs, l'impact du parallélisme sur les serveurs est moins clair : trop faible, il ne permet pas d'utiliser pleinement les ressources disponibles ; trop fort, il risque d'encombrer inutilement les serveurs de requêtes en attente. Nous étudions ce phénomène à l'aide d'un modèle de files d'attente où les requêtes arrivent selon un processus de Poisson et requièrent des traitements dont le volume suit une loi exponentielle. Chaque nouvelle requête est affectée à un certain nombre de serveurs, choisis de manière aléatoire, uniforme, et indépendante de l'état du système. Chaque serveur traite ses requêtes dans leur ordre d'arrivée. Nous montrons qu'il existe un degré de parallélisme qui minimise le nombre moyen de requêtes présentes dans chaque serveur. Ce degré optimal est de l'ordre de la racine carrée du nombre de serveurs pour une charge faible à modérée, et décroît jusqu'à deux à très forte charge.

Une version étendue de cet article est disponible en rapport de recherche [BCM17].

Mots-clefs : Cluster de serveurs, répartition de charge, parallélisme.

1 Introduction

Paralléliser les tâches dans les centres de calcul permet de réduire leur temps de traitement et d'optimiser l'utilisation des ressources. Même en l'absence de coût de parallélisation, par exemple lié à la communication entre les serveurs, il doit exister un degré de parallélisme optimal dans un tel système : trop faible, il ne permet pas d'utiliser pleinement les ressources disponibles ; trop fort, il risque d'encombrer inutilement les serveurs de requêtes. Les modèles de files d'attente traditionnels ne permettent pas de capturer ce phénomène [Kle75]. Nous nous intéressons ici à un modèle plus récent introduit par Gardner et al. [GHBSW+17] dans lequel les requêtes peuvent être parallélisées sur un nombre fixe de serveurs choisis de façon aléatoire, uniforme et indépendante de l'état du système, chaque serveur traitant ses requêtes dans leur ordre d'arrivée. Nous montrons qu'il existe un degré de parallélisme pour lequel le nombre moyen de requêtes sur chaque serveur est minimal. De manière assez surprenante, ce degré optimal est de l'ordre de la racine du nombre de serveurs à charge faible à modérée et décroît jusqu'à deux lorsque la charge est très forte.

2 Modèle

Soit un cluster de N serveurs, chacun ayant $\mu > 0$ pour capacité de traitement. Des requêtes arrivent dans le système selon un processus de Poisson d'intensité $N\lambda$; chaque requête est répartie sur d serveurs choisis au hasard de façon indépendante de l'état du système. La quantité de travail requise par chaque requête suit une loi exponentielle de moyenne unitaire. Les requêtes quittent le système à la fin de leur service.

Chaque serveur traite les requêtes qui lui ont été attribuées séquentiellement par ordre d'arrivée. Une requête peut être traitée efficacement en parallèle par plusieurs serveurs, de sorte que son taux de service est la somme des capacités des serveurs en train de la servir. Lorsqu'une requête est terminée, chacun de ses serveurs commence à traiter la requête suivante qui lui a été assignée. La charge totale de la file est notée $\rho = \frac{N\lambda}{N\mu} = \frac{\lambda}{\mu}$. Comme tous les serveurs sont interchangeables, ρ donne aussi la charge de chaque serveur.

† Les auteurs sont membres du LINCS, voir `http://www.lincs.fr`

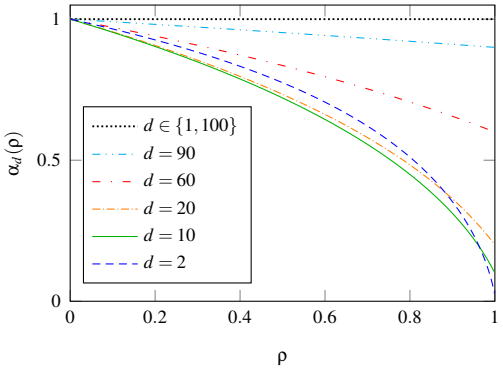


FIGURE 1: Nombre moyen normalisé de requêtes à traiter par chaque serveur en fonction de la charge ρ ($N = 100$).

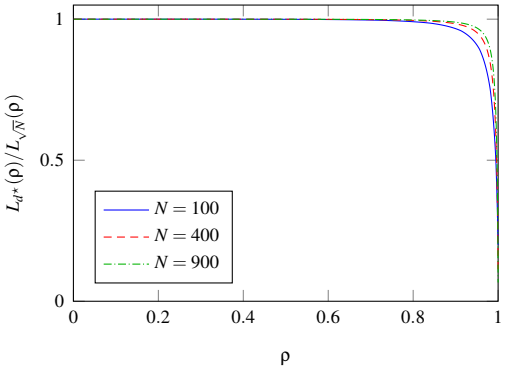


FIGURE 2: Optimalité du choix $d = \sqrt{N}$ en fonction de la charge ρ.

Ce système peut être vu comme une file d'attente multi-classe multi-serveur, chaque classe de requêtes correspondant à l'une des $\binom{N}{d}$ attributions possibles. Sans grande surprise, la file est stable si et seulement si $\rho < 1$ (cf. [GHBSW$^+$17]), ce que nous supposerons vérifié dans la suite. La file est décrite par un état agrégé (x,n), où x est le nombre de requêtes présentes et n le nombre de serveurs actifs (c'est-à-dire en train de traiter une requête). Cet état définit un processus stochastique, en général non markovien, sur

$$\mathcal{S} = \{(0,0)\} \cup \left\{(x,n) \in \mathbb{N}^2 : \ x \geq 1 \text{ et } d \leq n \leq \min(N, dx)\right\}.$$

On note π sa distribution à l'état stationnaire. Gardner *et al.* ont prouvé dans [GHBSW$^+$17, Théorème 4] que π satisfait la récurrence suivante : pour chaque $(x,n) \in \mathcal{S} \setminus \{(0,0)\}$, on a

$$\pi(x,n) = \frac{N\rho}{n} \sum_{\ell=0}^{d} \frac{\binom{n-\ell}{d-\ell}\binom{N-n+\ell}{\ell}}{\binom{N}{d}} \pi(x-1, n-\ell), \quad \text{avec la convention } \pi(x,n) = 0 \text{ si } (x,n) \notin \mathcal{S}. \tag{1}$$

Par la suite, on note $(\mathbf{X}, \mathbf{N})$ un couple de variables aléatoires suivant cette loi de probabilité.

3 Taille moyenne de la file d'attente par serveur

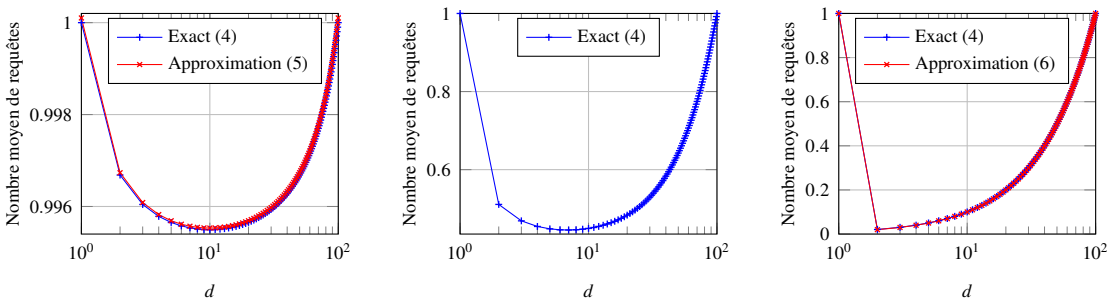
La formule suivante a été prouvée dans [GHBSW$^+$17, Théorème 1] :

$$\mathbb{E}(\mathbf{X}) = \sum_{n=d}^{N} \frac{\rho}{\frac{\binom{N-1}{d-1}}{\binom{n-1}{d-1}} - \rho}. \tag{2}$$

On note $L_d(\rho)$ le nombre moyen de requêtes sur chaque serveur. Tous les serveurs sont interchangeables et chaque nouvelle requête est répliquée sur d serveurs distincts, de sorte que

$$L_d(\rho) = \frac{d\mathbb{E}(\mathbf{X})}{N} = \frac{d}{N} \sum_{n=d}^{N} \frac{\rho}{\frac{\binom{N-1}{d-1}}{\binom{n-1}{d-1}} - \rho}. \tag{3}$$

Dans les cas particuliers où $d = 1$ et $d = N$, on obtient le nombre moyen $\rho/(1-\rho)$ de requêtes dans une file d'attente $M/M/1$ sous la charge ρ. Lorsque l'on regarde l'évolution de la file d'attente de chaque serveur, les deux situations sont effectivement équivalentes à des échelles de temps différentes. Dans le premier cas, on a N files $M/M/1$ indépendantes : chaque serveur voit arriver les requêtes au taux λ et les traite à vitesse μ selon la discipline FIFO. Dans le second cas, on a une unique file $M/M/1$: le taux d'arrivée des requêtes à chaque serveur est $N\lambda$ et celles-ci sont servies par ordre d'arrivée à vitesse $N\mu$. Il est montré dans [BCM17] que ces deux configurations constituent un pire cas pour le nombre moyen de requêtes par serveur.



(a) $\rho = 0.01$ (b) $\rho = 0.8$ (c) $\rho = 0.9999$

FIGURE 3: $\alpha_d(\rho)$ en fonction de d, pour différentes valeurs de ρ ($N = 100$).

Notre objectif est de minimiser l'encombrement moyen $L_d(\rho)$ et de quantifier le gain par rapport au pire cas $d = 1$, sans parallélisme. On s'intéresse donc à la quantité suivante, comprise entre 0 et 1 :

$$\alpha_d(\rho) \equiv \frac{L_d(\rho)}{L_1(\rho)} = \frac{\frac{d\mathbb{E}(\mathbf{X})}{N}}{\frac{\rho}{1-\rho}} = \frac{d}{N}\sum_{n=d}^{N} \frac{1-\rho}{\frac{\binom{N-1}{d-1}}{\binom{n-1}{d-1}} - \rho}. \tag{4}$$

L'évolution de $\alpha_d(\rho)$ en fonction de la charge ρ pour $N = 100$ serveurs est représentée sur la Figure 1 pour différentes valeurs de d. On observe tout d'abord que, exceptées les valeurs extrêmes $d = 1$ et $d = 100$, le parallélisme est d'autant plus efficace que la charge est forte. On voit également que le choix $d = 10$ est optimal sauf pour une charge ρ proche de 1, où $d = 2$ devient meilleur.

Afin de mieux comprendre le phénomène, la Figure 3 étudie trois « tranches » de la Figure 1, à savoir la valeur de $\alpha_d(\rho)$ en fonction de d pour une charge ρ faible, forte et extrême. À charge faible (Figure 3(a)), on observe une certaine symétrie de la courbe, en échelle logarithmique pour d, qui amène à penser que le choix $d = \sqrt{N}$ est optimal. Cette symétrie se brise lentement quand ρ augmente, et à forte charge (Figure 3(b)), le choix $d = \sqrt{N}$ n'est plus optimal, même s'il reste assez efficace . À charge extrême (Figure 3(c)), la symétrie est complètement brisée : la courbe est croissante pour $d \geq 2$.

Pour montrer l'intérêt de choisir pour d la racine carrée du nombre de serveurs, nous avons calculé à N et ρ fixés la valeur $d^\star$ de d qui minimise $L_d(\rho)$ et déduit l'optimalité relative $L_{d^\star}(\rho)/L_{\sqrt{N}}(\rho)$ du choix $d = \sqrt{N}$. Les résultats, reportés sur la Figure 2, confirment une quasi-optimalité de $d = \sqrt{N}$ sauf à très forte charge. On peut également remarquer que la baisse d'optimalité semble retardée pour N grand.

Nous allons maintenant prouver l'optimalité de $d = \sqrt{N}$ à faible charge et de $d = 2$ à forte charge, la quasi-optimalité de $d = \sqrt{N}$ à charge modérée restant pour l'instant une conjecture, même si elle est mise en évidence par les évaluations numériques.

4 Étude à faible charge

À faible charge, la performance du système est dictée par ce qu'il se passe en présence d'une ou deux requêtes. Celle qui est en tête de file est systématiquement en service sur d serveurs ; la seconde, si elle est présente, est en service sur $d-k$ serveurs, où k est le nombre de *collisions*, c'est-à-dire de serveurs qui sont attribués aux deux requêtes. Il y a en moyenne d^2/N collisions, puisque chacun des d serveurs attribués à la seconde requête peut avoir déjà été attribué à la première avec probabilité d/N. Intuitivement, on comprend que la valeur $d = \sqrt{N}$ marque une transition : si $d \ll \sqrt{N}$, il y a peu de collisions mais le parallélisme est faible ; si $d \gg \sqrt{N}$, les collisions, nombreuses, ne sont pas compensées par le gain du parallélisme. Le résultat suivant permet de formaliser cette intuition.

Proposition 1. *Pour* $1 \leq d \leq N$, *on a*

$$\alpha_d(\rho) \simeq \frac{1-\rho}{1 - \frac{\rho}{2-\left(\frac{1}{d}+\frac{d}{N}\right)}} \text{ lorsque } \rho \to 0. \tag{5}$$

La résolution de (5) montre que la valeur de d qui minimise $\alpha_d(\rho)$ est bien $\sqrt{N}$. La validité de l'approximation peut être observée sur la Figure 3(a) pour $N = 100$ et $\rho = 0.01$. On donne ici une idée de la preuve de (5), la preuve complète étant disponible dans [BCM17].

Idée de la preuve. On fait tendre $N\rho/d$ vers 0. Quand $N\rho/d$ est petit, on peut ignorer les états contenant plus de deux requêtes car leur impact dans $\alpha_d(\rho)$, en $O((N\rho/d)^2)$, est négligeable devant le reste. Plus précisément, en développant puis en simplifiant l'expression (4) de $\alpha_d(\rho)$ en fonction de $\mathbb{E}(\mathbf{X})$, on obtient

$$\alpha_d(\rho) = (1-\rho)\frac{1 + \frac{N\rho}{d}\sum_{k=0}^{d} \frac{1}{1-\frac{k}{2d}} p_k + O\left(\left(\frac{N\rho}{d}\right)^2\right)}{1 + \frac{N\rho}{d} + O\left(\left(\frac{N\rho}{d}\right)^2\right)},$$

où $p_k = \binom{d}{k}\binom{N-d}{d-k}/\binom{N}{d}$, pour $k = 0, 1, \ldots, d$, donne la probabilité qu'il y ait k collisions lorsque l'on tire d serveurs parmi N dont d sont déjà occupés (loi hypergéométrique de paramètres N, d, d). Il reste à donner une approximation de la somme où intervient cette loi. On distingue trois régimes selon les valeurs de d : d petit devant $\sqrt{N}$, d de l'ordre de $\sqrt{N}$ et d proche de N (ou plus précisément, $N-d$ petit devant $\sqrt{N}$). Dans les trois cas, on montre que

$$\sum_{k=0}^{d} \frac{1}{1-\frac{k}{2d}} p_k \simeq 1 + \frac{d}{N}\frac{1}{2-\frac{1}{d}-\frac{d}{N}}, \text{ dont on déduit le résultat.}$$

□

5 Étude à forte charge

Le comportement à forte charge est décrit par la proposition suivante.

Proposition 2. *Pour $2 \leq d \leq N$, on a*

$$\alpha_d(\rho) \simeq \frac{d}{N} \text{ lorsque } \rho \to 1. \tag{6}$$

En se souvenant que $\alpha_1(\rho) = 1$, on en déduit que le degré de parallélisme qui minimise le nombre moyen de requêtes par serveur à forte charge est $d = 2$. La validité de (6) est illustrée sur la Figure 3(c) pour $N = 100$ et $\rho = 1 - 1/N^2 = 0.9999$. L'idée de la preuve, donnée dans [BCM17], repose sur un développement limité de $\alpha_d(\rho)$ quand ρ tend vers 1. Dès que $d \geq 2$, le terme dominant dans (2) est $\rho/(1-\rho)$, correspondant à $n = N$. Le nombre moyen de requêtes *dans le cluster* s'approche ainsi de $\rho/(1-\rho)$ à très forte charge. En revenant à $\alpha_d(\rho)$, on obtient le résultat.

6 Conclusion

Nous avons étudié un modèle de cluster de serveurs avec traitement des requêtes en parallèle et montré qu'il existe un degré de parallélisme qui minimise le nombre moyen de requêtes sur chaque serveur. Ce degré optimal est de l'ordre de la racine du nombre de serveurs à faible charge et décroît jusqu'à 2 à forte charge. Dans les travaux futurs, nous souhaitons prouver les intuitions données par les résultats numériques et étudier la sensibilité du degré optimal à la loi de la taille des requêtes.

Références

[BCM17] T. Bonald, C. Comte, and F. Mathieu. À la racine du parallélisme. Research report, Telecom ParisTech, 2017. `https://hal.inria.fr/hal-01476889`.

[GHBSW+17] K. Gardner, M. Harchol-Balter, A. Scheller-Wolf, M. Velednitsky, and S. Zbarsky. Redundancy-d : The power of d choices for redundancy. *Operations Research*, 2017.

[Kle75] L. Kleinrock. *Queueing Systems, Volume I : Theory*. Wiley Interscience, New York, 1975.