



HAL
open science

Compromis entre énergie et performance dans le cloud

Mohamed Mehdi Kandi, Farah Ait Salaht, Hind Castel-Taleb, Emmanuel Hyon

► **To cite this version:**

Mohamed Mehdi Kandi, Farah Ait Salaht, Hind Castel-Taleb, Emmanuel Hyon. Compromis entre énergie et performance dans le cloud. Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication, May 2017, Quiberon, France. hal-01517666

HAL Id: hal-01517666

<https://hal.science/hal-01517666v1>

Submitted on 3 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Compromis entre énergie et performance dans le cloud

M.Kandi,² F. Ait Salaht,^{1,3} H.Castel-Taleb,² and E.Hyon^{1 †}

¹Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 Paris UMR 7606, 4 place Jussieu 75005 Paris France

²SAMOVAR, Télécom SudParis, CNRS, Université Paris Saclay, 9 rue Charles Fourier-91011 Evry Cedex, France

³Crest-ENSAI, Rennes, France

Nous évaluons les performances et la consommation énergétique dans un cloud modélisé par un modèle de files d'attente multi serveurs. Nous cherchons à adapter les ressources en fonction de la demande. Nous travaillons sur des politiques de type hystérésis, dans lesquelles les serveurs sont activés et désactivés selon l'occupation de la file. Notre objectif est de calculer les séquences de seuils d'activation et de désactivation permettant d'optimiser à la fois les performances et l'énergie, afin d'établir un compromis entre les deux. Nous proposons d'analyser mathématiquement la chaîne de Markov en calculant la distribution stationnaire par une forme close en partant de coupes sur l'espace d'état et de manière récursive en fonction de la probabilité du premier état. Nous définissons ensuite une mesure globale qui s'écrit comme une fonction de coût sur cette distribution stationnaire prenant en compte à la fois l'énergie et les performances, et nous proposons des heuristiques afin de définir des séquences de seuils permettant de minimiser cette mesure globale.

Keywords: Cloud, Files d'attente, Politique hystérésis, Performances, Energie, Optimisation

1 Introduction

Nous proposons, dans ce travail, de modéliser un système de cloud par un modèle de file d'attente multi serveurs, où chaque serveur représente une machine virtuelle exécutant les tâches des utilisateurs, et nous voulons optimiser la consommation énergétique de ce cloud. L'amélioration de l'économie d'énergie dans les modèles de files d'attente passe par des politiques telles qu'allumer/éteindre les serveurs (état ON/OFF) [AELH05] ou par des politiques à hystérésis dans lesquelles l'activation et la désactivation de réserves de serveurs s'effectue lorsque l'occupation de la file atteint certains seuils précis [Mit13]. Le calcul des séquences de seuils optimaux dépend de la distribution stationnaire de la chaîne de Markov qui modélise ce système. Il est clair que lorsque le nombre de machines virtuelles devient important, comme c'est le cas pour un cloud, ce calcul devient complexe. Dans notre étude, nous proposons une activation par blocs des serveurs, avec un nombre de seuils plus grand que 1, contrairement à [Mit13], et nous résolvons la chaîne en généralisant la forme close de [LNT02] par élimination de certaines contraintes sur les seuils. Cette généralisation permet de calculer des séquences de seuils optimales par des heuristiques, pour une mesure prenant en compte à la fois les performances et la consommation énergétique.

2 Description du modèle

Nous supposons un système de file d'attente multi serveurs. Les requêtes arrivent, selon un processus de Poisson de taux λ , dans une file d'attente. Elles sont servies par un ensemble de serveurs identiques dont le nombre total est C et dont le temps de service est exponentiel de taux μ . La capacité du système est de B clients. Tous les serveurs ne sont pas forcément actifs et on définit K niveaux de fonctionnement correspondant au nombre de serveurs, ou machines virtuelles (VM), actifs. Le nombre de serveurs actifs à un niveau $1 \leq k \leq K$ est noté S_k , avec $S_1 \leq S_2 \leq \dots \leq S_K = C$. On suppose que $S_1 \geq 1$ avec au moins un serveur actif. Le passage du niveau de fonctionnement k au niveau $k + 1$ permet d'activer un ou plusieurs serveurs,

[†]Ce travail est partiellement soutenu par le projet de recherche ANR-MARMOTE (ANR-12-MONU-0019)

faisant passer de S_k à S_{k+1} serveurs actifs, alors que le passage du niveau k au niveau $k - 1$ permet de désactiver certains serveurs faisant passer de S_k à S_{k-1} serveurs actifs. Notre modèle est caractérisé par des seuils d'activation, en fonction du nombre de clients dans le système, F_1, F_2, \dots, F_{K-1} , où F_k , pour $1 \leq k < K$, fait passer du niveau de fonctionnement k au niveau $k + 1$ et par des seuils de désactivation R_1, R_2, \dots, R_{K-1} , où R_k fait passer du niveau k au niveau $k - 1$. La désactivation se déroule à la fin d'un service et quand plusieurs serveurs sont désactivés simultanément les clients n'ayant pas fini leur service sont remis dans la file. On suppose aussi que $R_k < F_k$, que $F_1 < F_2 < \dots < F_{K-1}$ et que $R_1 < R_2 < \dots < R_{K-1}$, avec k dans $\{1 \dots K-1\}$. L'évolution du système est décrit par une chaîne de Markov à temps continu $\{X(t)\}_{t \geq 0}$, dont chaque état est représenté par le couple (m, k) où m est le nombre de clients dans le système, et k est le niveau de fonctionnement. Les équations d'évolution de la chaîne de Markov sont données comme suit :

- $(m, k) \rightarrow (\min\{B, m+1\}, k)$ avec un taux λ , si $m < F_k$;
- $\rightarrow (\min\{B, m+1\}, \min\{K, k+1\})$, avec un taux λ , si $m = F_k$;
- $\rightarrow (\max\{0, m-1\}, k)$, avec un taux $\min\{S_k, m\} \cdot \mu$, si $m > R_{k-1} + 1$;
- $\rightarrow (\max\{0, m-1\}, \max\{1, k-1\})$, avec un taux $\min\{S_k, m\} \cdot \mu$, si $m = R_{k-1} + 1$.

Nous présentons Figure 1 un exemple d'une telle chaîne de Markov.

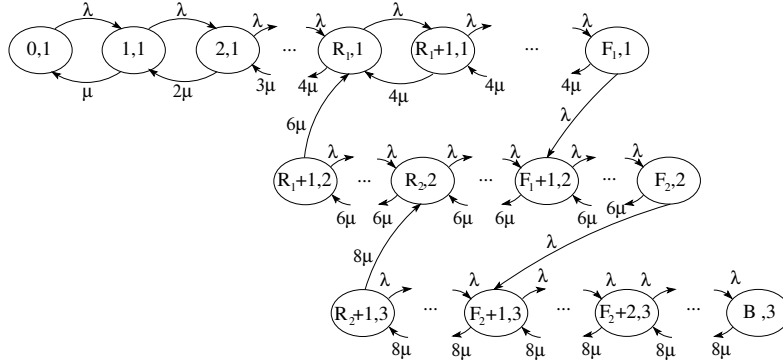


FIGURE 1: Graphe de transition de la chaîne avec $K = 3$, $S_1 = 4$, $S_2 = 6$, $S_3 = 8$, $R_1 + 1 \geq 6$, et $R_2 + 1 \geq 8$.

3 Forme close de la distribution stationnaire

Nous calculons la distribution stationnaire π par une forme close, déduite des équations de balance et des coupes sur l'espace d'état. Nous généralisons l'approche de [LNT02], car nous considérons les deux cas : $R_k \leq F_{k-1}$, et $R_k > F_{k-1}$, où $2 \leq k \leq K - 1$. L'idée générale consiste à calculer les probabilités stationnaires pour chaque état niveau par niveau, en commençant par le niveau 1. Ensuite la première probabilité du niveau 2, c'est à dire $\pi(R_1 + 1, 2)$, est calculée en fonction de $\pi(F_1, 1)$. Puis toutes les autres probabilités du niveau 2 sont calculées en fonction de la première probabilité du niveau 2 : $\pi(R_1 + 1, 2)$. L'approche peut être généralisée pour tout niveau $2 < k \leq K$.

Pour $k \in \{1 \dots K\}$, on note par $\mu_k = S_k \mu$, $\rho_k = \frac{\lambda}{\mu_k}$, et $\rho = \frac{\lambda}{\mu}$.

Lemme 1. Dans le niveau 1, le taux de service de l'état $(m, 1)$ est $m\mu$ si $1 \leq m < S_1$ et $S_1\mu$ si $m \geq S_1$. On en déduit la probabilité stationnaire de l'état $(m, 1)$:

$$\pi(m, 1) = \begin{cases} \frac{\rho^m}{m!} \pi(0, 1) & \text{si } 0 \leq m \leq S_1 ; \\ \rho_1^{m-S_1} \frac{\rho^{S_1}}{S_1!} \pi(0, 1) & \text{si } S_1 < m \leq R_1 ; \\ \frac{\rho^{S_1}}{S_1!} \left(\rho_1^{m-S_1} - \frac{\rho_1^{F_1-S_1+1} (1-\rho_1^{m-R_1})}{1-\rho_1^{F_1-R_1+1}} \right) \pi(0, 1) & \text{si } R_1 + 1 \leq m \leq F_1 . \end{cases} \quad (1)$$

Démonstration. Soit $0 \leq m < R_1$, nous effectuons une coupe autour de l'ensemble $\{(0, 1), \dots, (m, 1)\}$ (c'est à dire entre cet ensemble et le reste de l'espace d'état). Nous obtenons l'équation de balance $\mu(m+1)\pi(m+1, 1) = \lambda\pi(m, 1)$. On en déduit les deux premières lignes de l'équation (1).

Compromis entre énergie et performance dans le cloud

Soit $R_1 \leq m \leq F_1 - 1$, nous faisons une coupe autour de l'ensemble $\{(0, 1), \dots, (R_1, 1), \dots, (m, 1)\}$ et nous obtenons l'équation de balance $\mu_1 \pi(m+1, 1) + \mu_2 \pi(R_1+1, 2) = \lambda \pi(m, 1)$. On obtient, pour $R_1+1 \leq m \leq F_1$,

$$\pi(m, 1) = \rho_1^{m-R_1} \pi(R_1, 1) - \pi(R_1+1, 2) \frac{\rho_1}{\rho_2} \frac{1 - \rho_1^{m-R_1}}{1 - \rho_1}. \quad (2)$$

Si on remplace m par F_1 dans l'Equation (2), on obtient $\pi(F_1, 1)$ en fonction de $\pi(R_1, 1)$ et de $\pi(R_1+1, 2)$. En effectuant une coupe, entre les états du niveau 1 et les autres états, on obtient l'équation d'évolution suivante : $\lambda \pi(F_1, 1) = \mu_2 \pi(R_1+1, 2)$, dont on déduit $\pi(F_1, 1) = \frac{1}{\rho_2} \pi(R_1+1, 2)$. On veut, dans l'Equation (2), exprimer $\pi(R_1+1, 2)$ et $\pi(R_1, 1)$ en fonction de $\pi(0, 1)$. En posant $m = R_1$ dans la seconde ligne de (1) on obtient : $\pi(R_1, 1) = \rho_1^{R_1-S_1} \frac{\rho_1^{S_1}}{S_1!} \pi(0, 1)$ comme $\pi(R_1+1, 2)$ s'exprime en fonction de $\pi(R_1, 1)$, il vient $\pi(R_1+1, 2) = \frac{\rho_2 \rho_1^{F_1-S_1} (1-\rho_1)}{1-\rho_1} \frac{\rho_1^{S_1}}{S_1!} \pi(0, 1)$. On obtient la dernière ligne de l'équation (1). \square

Nous étudions le niveau k tel que $2 \leq k \leq K-1$. Le cas $R_k \leq F_{k-1}$ (correspondant au niveau 2 de la figure 1) pour $2 \leq k \leq K-1$ est traité dans [LNT02], nous proposons de donner les équations pour le cas $R_k > F_{k-1}$.

Lemme 2. Soit $k \in \{2 \dots K-1\}$. Nous supposons que $R_{k-1}+1 \geq S_k$. Le taux de service est $\min(R_{k-1}+1, S_k)$ et vaut $S_k \mu$. La probabilité stationnaire dans l'état (m, k) vaut

$$\pi(m, k) = \begin{cases} \frac{1 - \rho^{m-R_{k-1}}}{1 - \rho_k} \pi(R_{k-1} + 1, k) & \text{si } R_{k-1} + 2 \leq m \leq F_{k-1} + 1; \\ \frac{\rho_k^{m-F_{k-1}-1} - \rho_k^{m-R_{k-1}}}{1 - \rho_k} \pi(R_{k-1} + 1, k) & \text{si } F_{k-1} + 2 \leq m \leq R_k; \\ \frac{\rho_k^{m-F_{k-1}-1} - \rho_k^{m-R_{k-1}}}{1 - \rho_k} \pi(R_{k-1} + 1, k) - \frac{\rho_k}{\rho_{k+1}} \frac{1 - \rho_k^{m-R_k}}{1 - \rho_k} \pi(R_k + 1, k + 1) & \text{si } R_k + 1 \leq m \leq F_k. \end{cases}$$

Démonstration. Chacune des équations est obtenue à partir d'une coupe sur l'espace d'état. On peut voir que ces équations permettent d'exprimer la probabilité de l'état en fonction de la première probabilité du niveau k , c'est à dire de $\pi(R_{k-1}+1, k)$, qui s'exprime aussi en fonction de $\pi(F_{k-1}, k-1)$, par la coupe entre le niveau $k-1$ et le niveau k . Ainsi, on peut écrire toutes les probabilités en fonction de $\pi(0, 1)$, et par la condition de normalisation, on déduit les probabilités d'états stationnaire. \square

4 Résultats numériques

On définit la mesure globale suivante pour les performances et l'énergie :

$$\bar{C} = C_H \cdot \bar{N}_C + C_S \cdot \bar{N}_S + C_A \cdot \bar{N}_A + C_D \cdot \bar{N}_D + C_R \cdot \bar{N}_R,$$

où \bar{N}_C est le nombre moyen de clients, \bar{N}_R : le nombre moyen de clients perdus, \bar{N}_S : le nombre moyen de serveurs actifs, \bar{N}_A : le nombre moyen d'activations par unité de temps et \bar{N}_D : le nombre moyen de désactivation par unité de temps. Pour les coûts, nous définissons par C_H : le coût, par unité de temps, du maintien d'un client dans la file, C_S : le coût d'utilisation, par unité de temps, d'un serveur, C_A : le coût d'activation d'un serveur, C_D : le coût de désactivation et par C_R : le coût de perte d'un client.

Notre objectif dans un premier temps est de comparer la méthode de résolution développée ici avec les méthodes communément utilisées dans la littérature : les méthodes d'agrégation SCA [LG99] et LDQBD [Kha10] que nous avons adaptées à notre cas. Bien que les méthodes de résolutions SCA, LDQBD soient des méthodes numériques et que la nôtre soit sous forme close, l'avantage en précision de calcul n'est pas flagrant car l'écart de valeur sur le vecteur de distribution stationnaire entre les différentes méthodes n'excède pas 10^{-12} . La comparaison des temps de calcul est donnée dans le tableau 1. Nous observons clairement que la méthode de la forme close développée est la plus rapide et de manière importante pour les espaces d'états de grandes tailles où elle est entre 300 et 1000 fois plus rapide que les méthodes classiques.

Dans un second temps, nous allons nous intéresser au calcul des valeurs de seuils qui minimisent la mesure globale \bar{C} . Cette mesure globale est calculée pour une séquence de seuils donnée. Pour déterminer

Modèle	SCA	LDQBD	Forme close
K = 10, B = 750, (1271 états)	0.0406 s	0.0121 s	0.0091 s
K = 100, B = 7500, (13421 états)	0.9587 s	0.9889 s	0.0823 s
K = 1000, B = 75000, (134921 états)	307.54 s	1330.27 s	1.0561 s

TABLE 1: Comparaison des approches de résolution en termes de temps d'exécution (en secondes).

la séquence de seuils qui minimise la mesure, il faut donc parcourir l'espace des seuils et calculer celle-ci pour chacun d'eux. Nous avons adapté au cas hystérésis (avec deux types de seuils), les trois heuristiques de recherche locale à un seul seuil, proposées dans [KvH07]. Pour $B = 30$ et $K = 3$, nous avons effectué 200 tests. Ces heuristiques comme on peut le constater dans le tableau 2, ne donnent pas forcément la solution optimale (solution obtenue par énumération), mais ont des temps de calculs très courts.

-	Solutions optimales trouvées	Temps d'exécution moyen	Temps d'exécution maximal
Algorithme 1	188 sur 200 tests	0.441 sec	2.424 sec
Algorithme 2	191 sur 200 tests	0.386 sec	1.285 sec
Algorithme 3	173 sur 200 tests	0.119 sec	0.510 sec
Énumération de toutes les solutions	200 sur 200 tests	143.52 sec	382.75 sec

TABLE 2: Comparaison entre les heuristiques proposées et l'énumération de toutes les solutions

5 Conclusion

Nous avons développé dans ce travail une forme close pour un système de cloud modélisé par une file d'attente à hystérésis dont les activations et désactivations des machines virtuelles se font par bloc. Nous avons montré l'impact et l'intérêt que pouvait représenter cette méthode dans l'évaluation des performances et de la consommation d'énergétique. Nous avons ensuite adapté trois heuristiques de calcul de seuils qui minimisent le compromis entre performances et consommation d'énergie. Les travaux futurs viseront à améliorer ces heuristiques et développer d'autres algorithmes d'optimisation.

Références

- [AELH05] Jesus R Artalejo, Antonis Economou, and Maria Jesus Lopez-Herrero. Analysis of a multiserver queue with setup times. *Queueing Systems*, 51(1-2) :53–76, 2005.
- [Kha10] Jeffrey P. Kharoufeh. *Level-Dependent Quasi-Birth-and-Death Processes*. Wiley & Sons, 2010.
- [KvH07] A.A. Kranenburg and G.J. van Houtum. Cost optimization in the (S-1, S) lost sales inventory model with multiple demand classes. *Oper. Res. Lett.*, 35(4) :493–502, 2007.
- [LG99] John C.S. Lui and Leana Golubchik. Stochastic complement analysis of multi-server threshold queues with hysteresis. *Performance Evaluation*, 35(1) :19–48, 1999.
- [LNT02] Louis-Marie Le Ny and Bruno Tuffin. A simple analysis of heterogeneous multi-server threshold queues with hysteresis. In *Applied Telecommunication Symposium (ATS)*, 2002.
- [Mit13] Isi Mitrani. Managing performance and power consumption in a server farm. *Annals of Operations Research*, 202(1) :121–134, 2013.