Proposition de projet PAF 2019

TITRE : « **Optimisation de Pareto CPU-température pour l'affectation de machines virtuelles dans un datacenter** »

Encadrant : Maurice Gagnaire (I304)
Occurrence : 2019
Nombre d'étudiants minimum : 2
Nombre d'étudiants maximum : 4
Nombre d'instances : 1
Domaines : *Cloud Computing*, optimisation combinatoire, maths appliquées

1) Contexte du projet :

Le ***Cloud Computing*** consiste en la mise à disposition à la demande via l'Internet de ressources de calcul à partir de grands ***datacenters***. Une requête de calcul (ou ***job***) se caractérise par sa capacité **CPU** (en nombre de cycles d'horloge CPU par seconde), son espace mémoire **RAM** (en octets), son espace **disque** (en octets) et ses ressources logicielles. Un *datacenter* est structuré en baies de serveurs ou machines physiques (**PM**) régulièrement espacées dans un local climatisé. Tout ***job*** généré par un utilisateur nécessite l'activation d'une machine virtuelle (**VM**) sur une (ou plusieurs) des PMs du *datacenter*. De très nombreuses études ont été réalisées afin de définir des règles d'optimisation de l'utilisation des serveurs d'un *datacenter*. Cette optimisation repose sur l'algorithme combinatoire dit du ***bin packing*** à deux dimensions (CPU et RAM), la mémoire disque étant nettement moins onéreuse et étant beaucoup plus simple à gérer. Dans la pratique, la capacité CPU effectivement consommée par un job est sporadique. La probabilité pour que tous les jobs se partageant les cycles d'un même processeur aient besoin simultanément de consommer leur valeur crête est négligeable. C'est la raison pour laquelle la capacité CPU est une contrainte flexible. Ainsi, un *job* peut transitoirement utiliser une fraction de cycles CPU supérieure à celle effectivement réservée, si cela

n'impacte pas la qualité de service d'autres jobs partageant ce même processeur. Nous nous limiterons donc à vérifier que la capacité RAM est respectée. Un algorithme très simple en la matière de *bin packing* à deux dimensions sera présenté par l'encadrant en début de projet afin que celui-ci soit implémenté dans un simulateur (idéalement en matlab ou en langage C).

2) Les deux objectifs du projet:

2.1) Objectif #1 : mise en œuvre de l'algorithme du *bin packing* à deux dimensions pour placer les jobs clients sur les différents serveurs d'un *datacenter*.

Le bin packing CPU/RAM sur des les PMs peut s'avérer plus ou moins efficace selon la stratégie adoptée. Nous ferons l'hypothèse que les requêtes de calcul arrivent au portail Web du *datacenter* suivant un processus de Poisson de paramètre λ (en nombre de requêtes de calcul par seconde). L'espacement temporel moyen entre deux requêtes successives est donc $1/\lambda$. Le processus d'allocation des VMs aux PMs (algorithme du *bin packing*) est activé toutes les Δ secondes ($\Delta >> 1/\lambda$) pour être appliqué sur la fenêtre temporelle de durée Δ suivante. On fera l'hypothèse (réaliste) que le temps de traitement du *bin packing* est négligeable par rapport à $1/\lambda$. Ainsi, sur la **Figure 1,** toutes les requêtes arrivées sur la première période de durée Δ sont servies sur des serveurs disponibles du *datacenter* lors de fenêtre temporelle suivante.

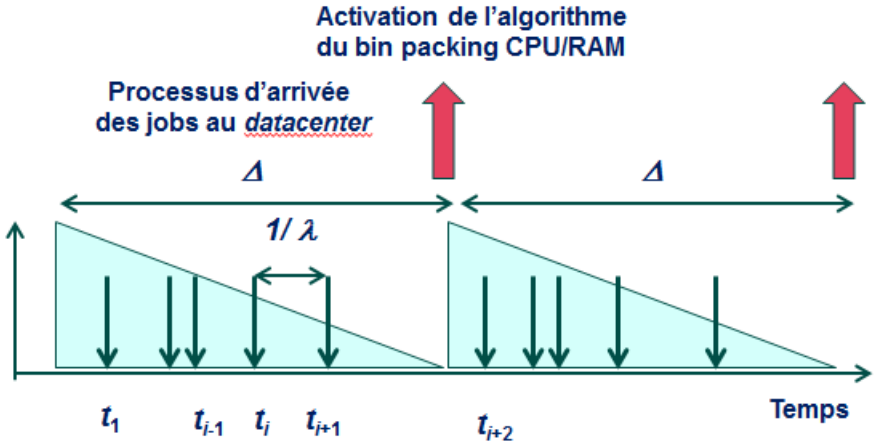

Figure 1. Processus de réservation de ressources CPU et RAM.

Chaque requête de calcul spécifie la durée d'utilisation CPU eu égard au nombre de cycles d'horloge CPU escomptés. Nous ferons l'hypothèse que cette durée de vie suit une loi uniforme sur l'intervalle [T_{min}, T_{max}]. Ainsi, après avoir accumulé toutes les requêtes de calcul pendant la fenêtre temporelle #*i de durée* Δ, il est possible de planifier l'utilisation partagée de chacun des processeurs nécessaires dans le datacenter sur la fenêtre suivante de même durée. Le résultat du *bin packing* obtenu est délivré à l'instant de début de la fenêtre temporelle #(i + 1). Il faut noter qu'il est possible que des résidus de requêtes censées être servies pendant la fenêtre #(i + 1) le soient dans des fenêtres ultérieures eu égard à l'inertie du système.

L'architecture type d'un datacenter sera assimilée à la configuration présentée en **Figure 2**.

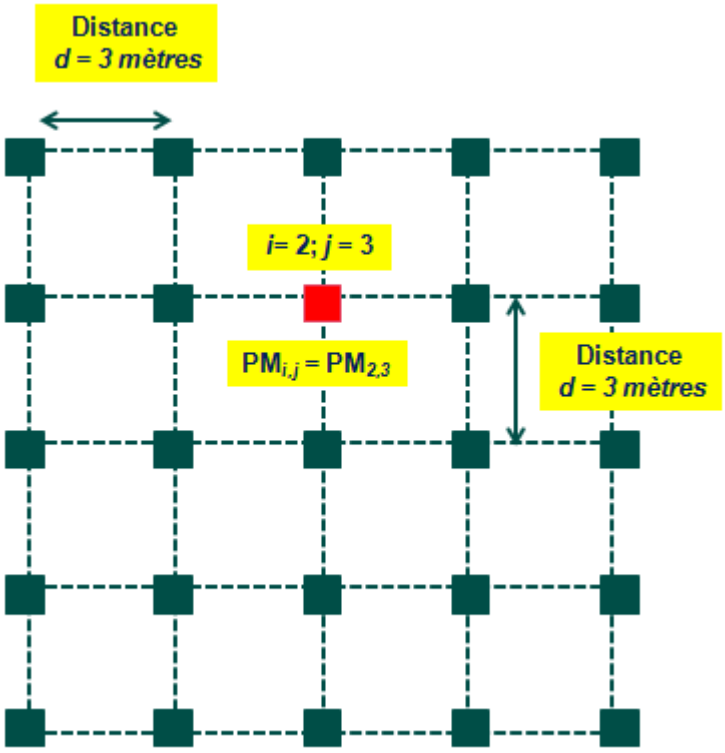

Figure 2. Configuration type d'un *datacenter*.

2.2) Objectif #2 : prise en compte des interactions thermiques entre serveurs sur la politique de placement des jobs.

3) Les hypothèses à prendre en compte

La très grande majorité des algorithmes de placement de VMs dans des PMs négligent l'impact de la **diffusion thermique** inhérente à la l'échauffement des serveurs. Il s'avère dans la pratique qu'un serveur atteignant une température de l'ordre de **70°C** est automatiquement déconnecté de l'alimentation électrique. A l'approche de telles températures, des requêtes de migration massives de VMs sont automatiquement activées en urgence vers des zones plus froides du *datacenter*. Les contraintes de bande passante du réseau *backplane* du *datacenter* sont telles qu'une telle opération est vouée à l'échec. L'expérience a montré en vraie grandeur, que le *datacenter* s'écroule en une vingtaine de minutes. Cela représente une lourde perte pour le **Cloud Service Provider** (**CSP**). Un tel événement s'est produit en 2017 dans l'un des *datacenters* du CSP français OVH. L'objectif et l'originalité de ce projet PAF est d'appréhender de façon simplifiée le problème de l'affectation de tâches de calcul sur un *datacenter* en pareille situation. Il s'agit de mettre en évidence les migrations à mettre en œuvre eu égard à l'évolution de la cartographie thermique du *datacenter* considéré. Nous faisons l'hypothèse que la capacité de calcul global du *datacenter* est connue et fixée telle que le décrit la **Figure 2**. La chaleur dissipée par une baie est proportionnelle au nombre des jobs actifs supportés à un instant donné.

Afin de bien cadrer les hypothèses de l'étude, trois types de *jobs* correspondant à des charges CPU de **20%, 40% ou 60%** seront considérées (on notera que si trois jobs de ce type sont affectés à un même processeur simultanément, le traitement des tâches considérées se fait obligatoirement plus lentement que souhaité. Une telle flexibilité ne peut pas s'appliquer à la capacité RAM cumulée sur un même serveur. En l'état de la technologie, on considérera des serveurs « lamme » Intel Xeon équipés de CPUs à **1,5 GHz** et d'une mémoire RAM de **32 Gbytes**. La consommation énergétique unitaire d'un serveur Intel

Xeon avec les caractéristiques nommées plus haut est à plein régime égale à **8 Watts**.

4) Les bases théoriques de la diffusion thermique dans un datacenter et leur exploitation à l'échelle de ce projet

La modélisation des phénomènes d'interférence thermique dans un *datacenter* repose sur des bases d'analyse numérique et de physique (thermodynamique, mécanique des fluides) complexes qui vont très au-delà des objectifs de ce projet On pourra néanmoins parcourir quelques une des références jointes dans la bibliographie ([1] , [2], [3], [4]).

Un modèle simplifié de diffusion thermique

Nous faisons l'hypothèse que la distribution de la chaleur dégagée par une baie de serveur suit une loi Gaussienne tel que décrit dans la Figure 3 ci-dessous.

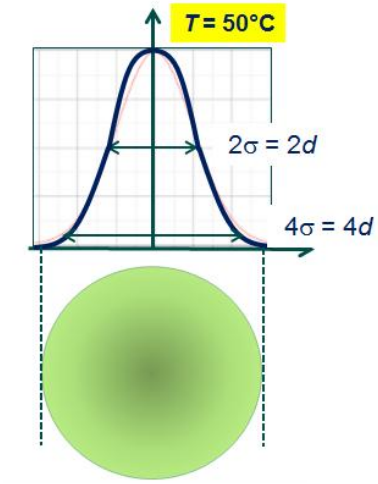

Figure 3 : diagramme de diffusion thermique d'un serveur

Plus la charge CPU est élevée, plus haute est la température au dessus de la baie et plus grand est son rayon d'influence sur les baies voisines. On considérera que lorsque plusieurs disques du type représenté en Figure 3 se

chevauchent, les températures associées se cumulent. La connaissance théorique de cette cartographie thermique du *datacenter* peut permettre de concevoir de nouvelles règles d'affectation des VMs au PMs afin d'optimiser l'efficacité globale du *datacenter*. La **Figure 4** illustre un exemple de telles interactions thermiques en un point du datacenter. En une point, la température mesurée est la somme des gaussiennes thermiques des baies environnantes. Rappelons que la largeur de ces gaussiennes dépend directement de la charge CPU des baies actives dans le datacenter.

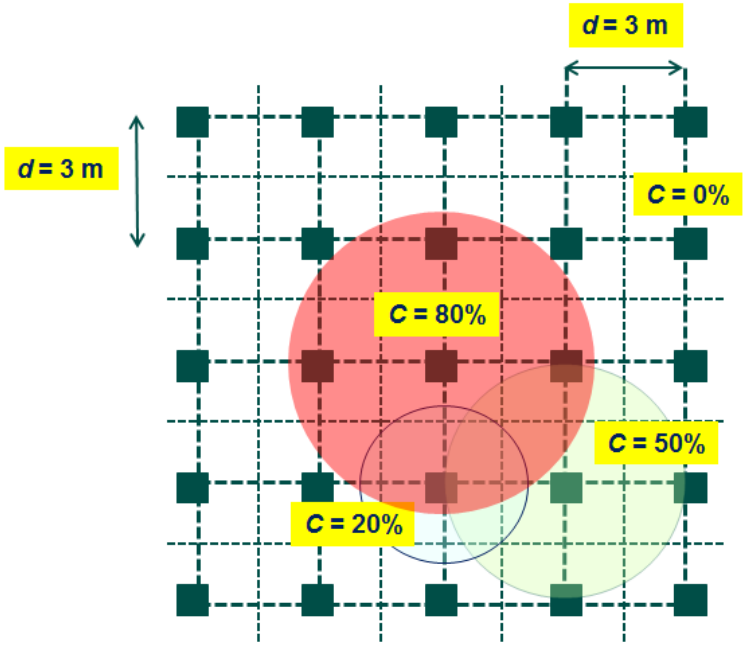

Figure 4. Exemple de *datacenter* composé de 25 baies de serveurs dont trois interfèrent thermiquement.

La dimension « **optimisation de Pareto** » de cette étude résulte du compromis entre la maximision de l'usage CPU de chaque baie, et d'autre part, la minimisation des interférences thermiques interbaies dans le *datacenter*. On peut approximer la chaleur dissipée par un processeur comme étant proportionnelle au cube f^{+3} de sa fréquence opérationnelle f. Cette chaleur décroit sous forme de gaussienne autour de chaque baie active. **L'objectif à satisfaire est donc, pour plusieurs scénarios de charge donnés, d'affecter le plus grand nombre de tâches de calcul aux différentes baies de serveurs (les**

25 points de la grille représentée en Figure 4) tout en garantissant qu'aucune baie n'atteigne la température fatidique des 70°C.

5) Références

[1] Cisco white paper, "Unified Computing System Site Planning Guide: Data Center Power and Cooling", 2017.

[2] Z. Song, X. Zhang, C. Eriksson, "Data Center Energy and Cost Saving Evaluation", 7th International Conference on Applied Energy, 2015.

[3] Vanessa López ⇑, Hendrik F. Hamann, « Heat transfer modeling in data centers », International Journal of Heat and Mass Transfer, International Journal of Heat and Mass Transfer, 2011.

[4] Behzad Norouzi-Khangah, Mohammd Bagher, Mohammad Sadeghi-Azad, Seyed Morteza Hoseyni, Seyed Mohsen Hoseyni, "Performance assessment of cooling systems in data centers: methodology and application of a new thermal metric", Elsevier journal on Cases Studies in Thermal Engineering, 2016.