

Ordonnancement sous contraintes de Qualité de Service dans les Clouds

GUÉROUT Tom



**DA COSTA Georges
(SEPIA)**



**MONTEIL Thierry
(SARA)**

14/09/2015

Profil

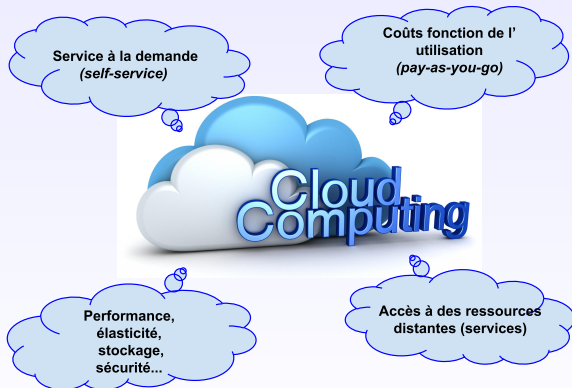
Parcours : Laboratoires LAAS et IRIT à Toulouse

- 2011-2014 : **Doctorat**
“Ordonnancement sous contraintes de Qualité de Service dans les Clouds”
- 2015 : **Post-Doctorat**
“Gestion de Qualité de Service pour l’Internet des Objets”
- *2016* : **Chargé de Recherche** au LAAS

Domaines de recherche

Cloud-Computing, Internet des Objets, Qualité de Service, Études énergétiques, Ordonnancement, Optimisation multi-objectifs, Simulation

Thèse : Contexte de recherche



Contributions de Thèse

Modélisation QoS Cloud

- * Définition et catégorisation de paramètres de QoS Cloud
- * **Élaboration de métriques mesurables de paramètres non-fonctionnels de QoS pour les SLA des fournisseurs de Cloud**

Études de consommations énergétiques

- * Étude de l'outil de **gestion de fréquences CPU du noyau Linux : le DVFS, et implémentation au sein du simulateur CloudSim**
- * Étude de modèles énergétiques
- * **Validation de simulations avec DVFS par comparaisons avec des expérimentations réelles (Grid'5000, Super-calculateur)**

Optimisation multi-objectifs QoS Cloud

- * **Problème (NP-Complet) de placement de machines virtuelles sous contraintes de QoS**
- * Résolution par une heuristique (Algorithme Génétique)
- * Évaluation de la qualité d'optimisation

Simulation d'ordonnancement Cloud

- * **Ordonnancement (ré-allocation) multi-objectifs**
- * Analyse de l'impact de l'optimisation par le GA des algorithmes gloutons

Modélisation QoS Cloud

Catégories de QoS :

- PERFORMANCE
- SÛRETÉ DE FONCTIONNEMENT
- SÉCURITÉ & DONNÉES
- COÛT

Avec comme buts majeurs de :

- Satisfaire l'utilisateur par de bonnes conditions d'utilisation
- Tirer profit de ces services

Compromis entre la QoS proposée à l'utilisateur et les coûts engendrés

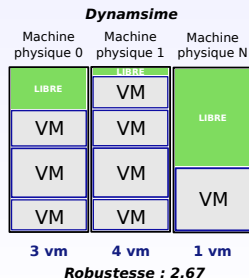
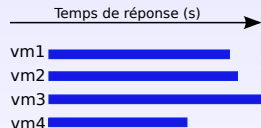
Modélisation QoS Cloud

Catégories de QoS :

- **PERFORMANCE :**
Temps d'exécution, Latence, Débit,
Temps de réponse
- **SÛRETÉ DE FONCTIONNEMENT :**
Robustesse et Dynamisme
- **SÉCURITÉ & DONNÉES :**
Authentification, Autorisation, Intégrité,
Confidentialité, Responsabilisation, Traçabilité,
Cryptage, Cycle de vie des données, Non Répudiation
- **COÛT :**
Coût de service, Empreinte carbone,
Coût énergétique :

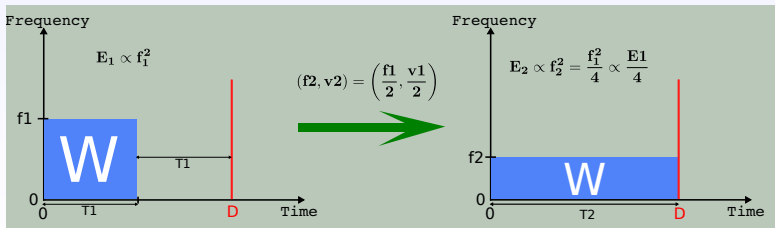
$$E = \sum_{i=1}^{n_H} P(F_i)^{h_i} \times T_{rep}^{h,k}$$

$$P(F_i) = \alpha (P_{full}(F_i) - P_{idle}(F_i)) + P_{idle}(F_i)$$



Leviers énergétiques

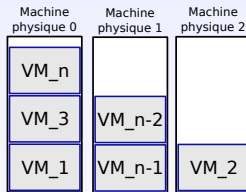
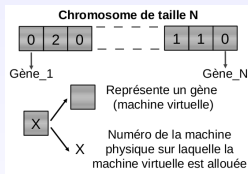
- ON/OFF machines physiques
- Consolidation VM (Migrations)
- Reconfiguration de la capacité de calcul des VM
- Utilisation du **DVFS** (Dynamic Voltage & Frequency Scaling)
 - PERFORMANCE : Fréquence maximum fixée
 - POWERSAVE : Fréquence minimum fixée
 - USERSPACE : Fréquence choisie fixée
 - ON-DEMAND : Décision dynamique avec 1 seuil + 1 délai
 - CONSERVATIVE : Décision dynamique avec 2 seuils



Optimisation multi-critères : Algorithme Génétique

Modélisation :

- Un chromosome \Leftrightarrow une solution de placement
- Un gène \Leftrightarrow une machine virtuelle
- Valeur d'un gène \Leftrightarrow numéro de la machine physique



Fonction objectif (valeur de Fitness) :

$$F_{obj} = \alpha_1 \times n(E) + \alpha_2 \times n(RespT) + \alpha_3 \times n(Rob) - \alpha_4 \times n(Dyn)$$

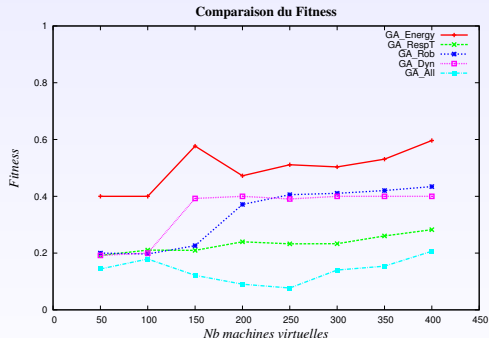
Fonction de normalisation "Centrer-Réduire" : $n(x) = \frac{x-\mu}{\sigma}$

\Rightarrow Valeurs de fitness comparables

Méta-heuristique : Algorithme Génétique

TABLE: Coefficient values of the 5 versions of the GA

GA Name	Coefficients applied to metrics			
	Energy	RespT time	Rob	Dyn
GA_All	1	1	1	1
GA_Energy	1	0	0	0
GA_RespT	0	1	0	0
GA_Rob	0	0	1	0
GA_Dyn	0	0	0	1



Ordonnancement par Simulation

Contribution au simulateur :

- Intégration de nouvelles fonctionnalités dans **CloudSim**
- Collaboration avec l'équipe de Raj. Buyya de Melbourne¹
- Validation par comparaison avec des expés réelles (grid'5000, Hyperion)
- Simulation d'ordonnancement périodique afin d'analyser l'évolution des métriques de QoS au cours du temps²

Condition de simulation :

- Instance du problème : 110 / 400
- Même caractéristiques que pour l'étude du GA
- Ré-Allocations périodiques chaque 20 secondes ⇒ migrations

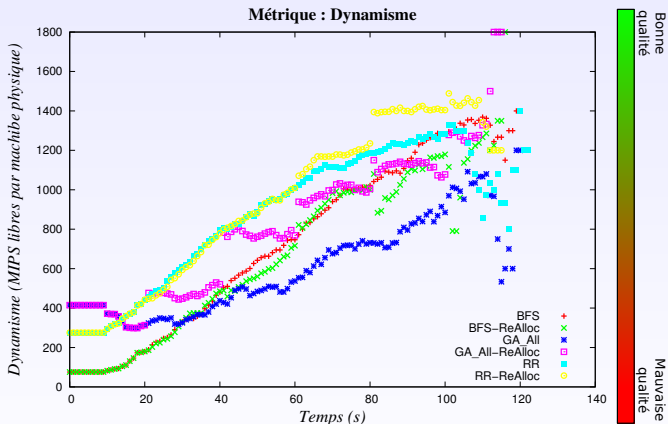
Algorithmes de placement utilisés :

- Allocation avec les différentes versions du GA
- Allocation avec des algorithmes gloutons :
Best-Fit Sorted (BFS) & Round-Robin (RR)

1. Journal : **Energy-aware simulation with DVFS**, *Simulation Modelling Practice and Theory*, 2013

2. Journal : **Quality of service modeling for green scheduling in Clouds**, *SUSCOM*, 2014

Résultat - Dynamisme



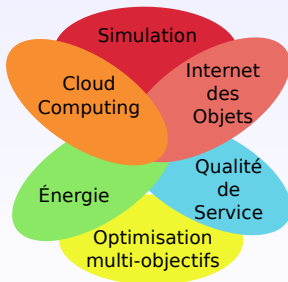
Courbes GA/BFS/RR : Allocation unique à $t=0$

Courbes GA_All-ReAlloc/BFS-ReAlloc/RR-ReAlloc : Allocation à $t=0$ puis ré-allocation chaque 20 secondes

GUÉROUT Tom

tguerout@laas.fr

http ://www.laas.fr/~tguerout

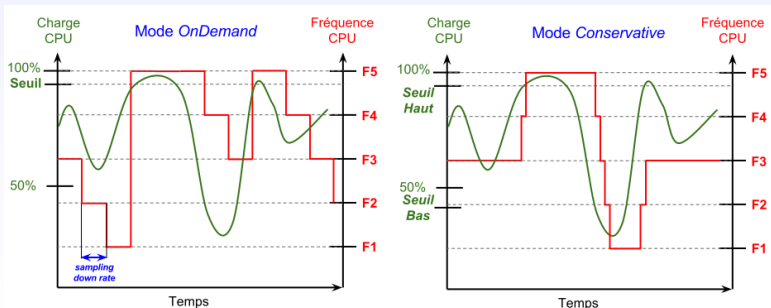


DVFS dans le noyau Linux

5 modes de fonctionnement : 3 statiques, 2 dynamiques

- PERFORMANCE : Fréquence maximum fixée
- POWERSAVE : Fréquence minimum fixée
- USERSPACE : Fréquence choisie fixée
- ON-DEMAND : Décision dynamique avec 1 seuil + 1 délai
- CONSERVATIVE : Décision dynamique avec 2 seuils

Prise de décision de changement de fréquence chaque 10ms



Pourquoi changer la fréquence des CPU ?

Puissance dynamique délivrée par un composant CMOS :

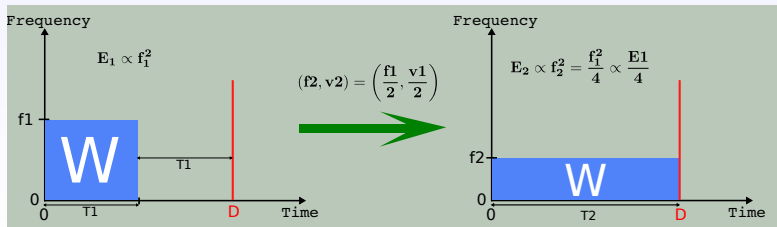
$$P_{cmos} = C_{eff} \times V^2 \times f$$

avec, C_{eff} la capacitance effective *, V le voltage et f la fréquence

* grandeur physique : capacité d'un composant à s'opposer au changement de voltage entre ses bornes

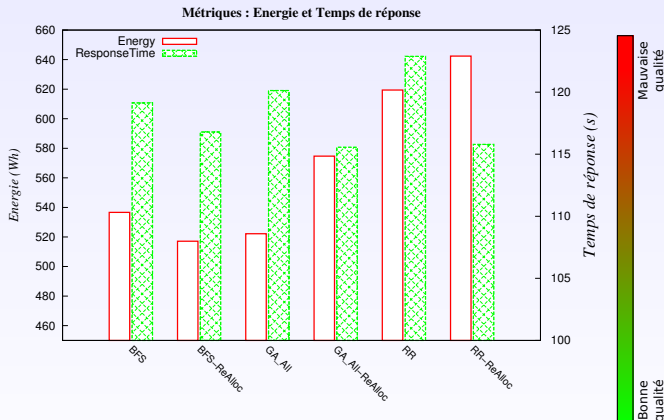
Énergie dépensée par tâche :

$$E = P * T \propto V^2, \text{ avec } V \propto f, \text{ alors } E \propto f^2$$



Fréquence divisée par 2 → Énergie divisée par 4

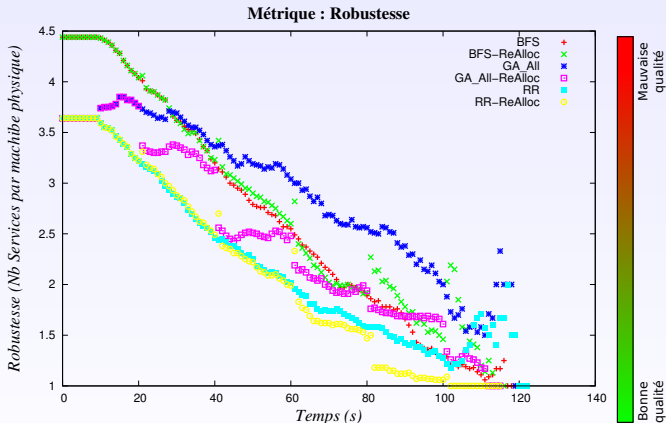
Résultat - Énergie / Temps de réponse



Courbes GA/BFS/RR : Allocation unique à $t=0$

Courbes GA_All-ReAlloc/BFS-ReAlloc/RR-ReAlloc : Allocation à $t=0$ puis ré-allocation chaque 20 secondes

Résultat - Robustesse



Courbes GA/BFS/RR : Allocation unique à t=0

Courbes GA_All-ReAlloc/BFS-ReAlloc/RR-ReAlloc : Allocation à t=0 puis ré-allocation chaque 20 secondes