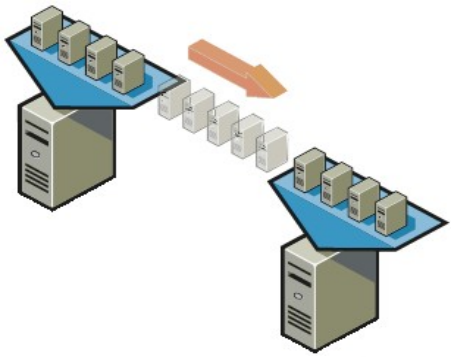
Ordonnancement contrôlé de migrations à chaud

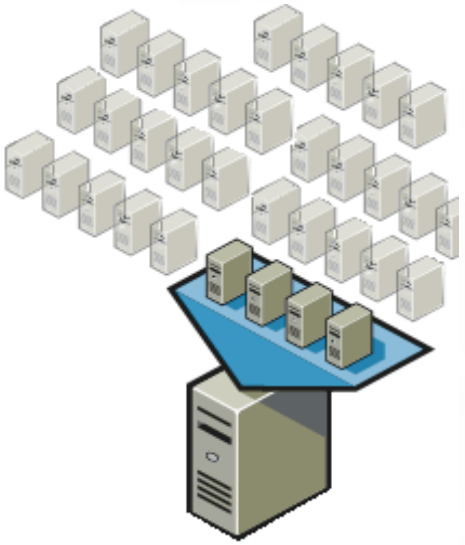
Vincent Kherbache, *Fabien Hermenier*,
Eric Madelaine

La migration à chaud

- Principe
  - Déplacer une VM en cours de fonctionnement entre différents serveurs physique



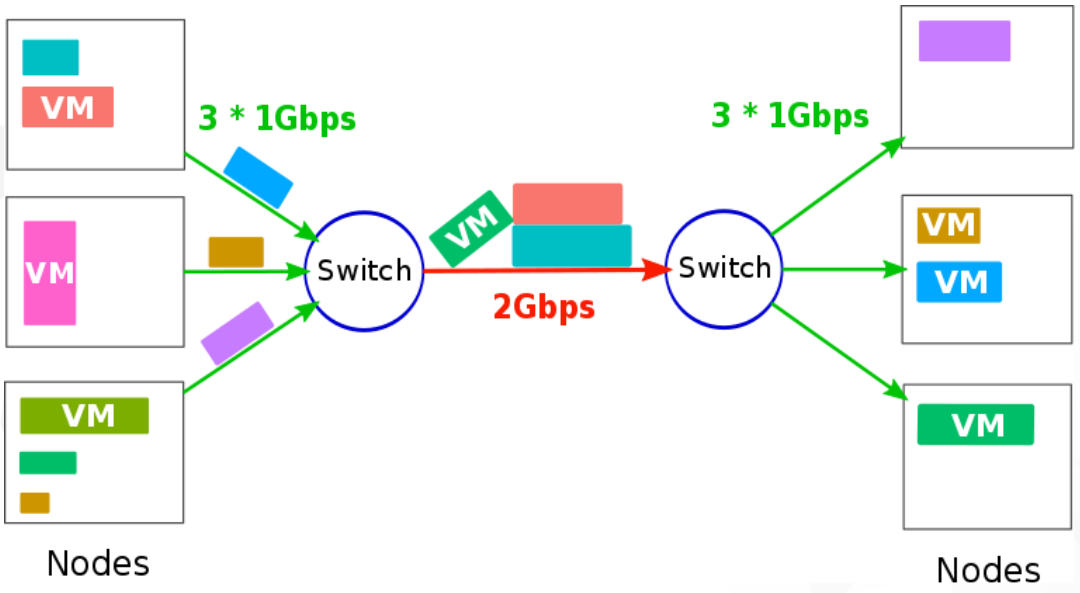
- Usages
  - Répartition / gestion de charge
  - Tâches de maintenance sur serveurs de production
  - Réduction de la consommation énergétique



Migrations multiples

- Pour bénéficier des avantages d'un nouveau placement de VM, il faut pouvoir migrer le plus rapidement possible.

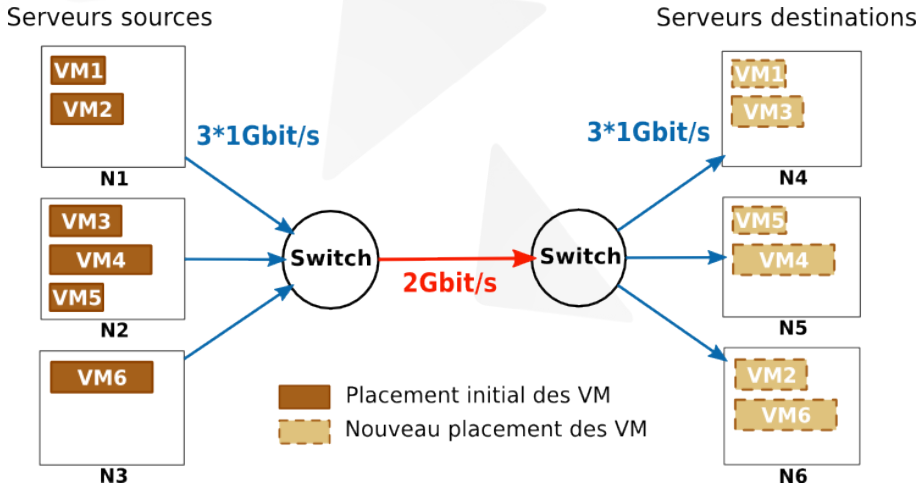
On ne doit pas saturer le réseau



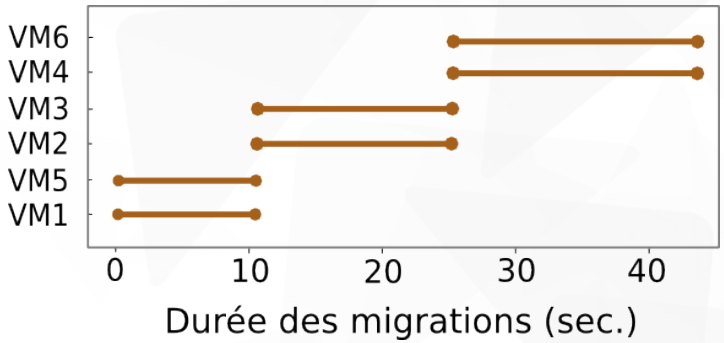
Ordonnancement de migrations

Déterminer pour chaque migration :

- La bande passante à allouer
- Sa durée théorique
- Le moment où la lancer



- Parallélisme dépendant de la topologie



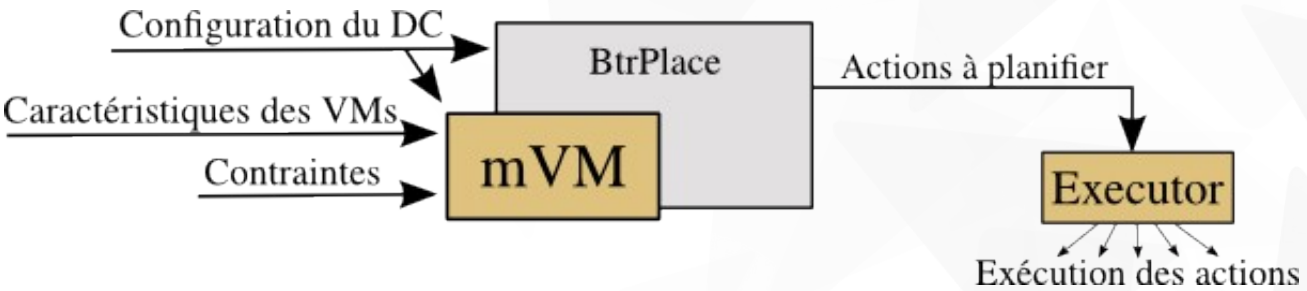
État de l'art

- Solutions proposées : [Entropy, BtrPlace, Memory Buddies, CloudSim, ..]
  - Réseaux non-bloquant
  - Workload ignorées
  - Parallélisation abusive ou inadaptée
- Conséquences :
  - Sous-estimation des durées
  - Migrations inutilement longues
  - Réduction des performances des VM

Solution

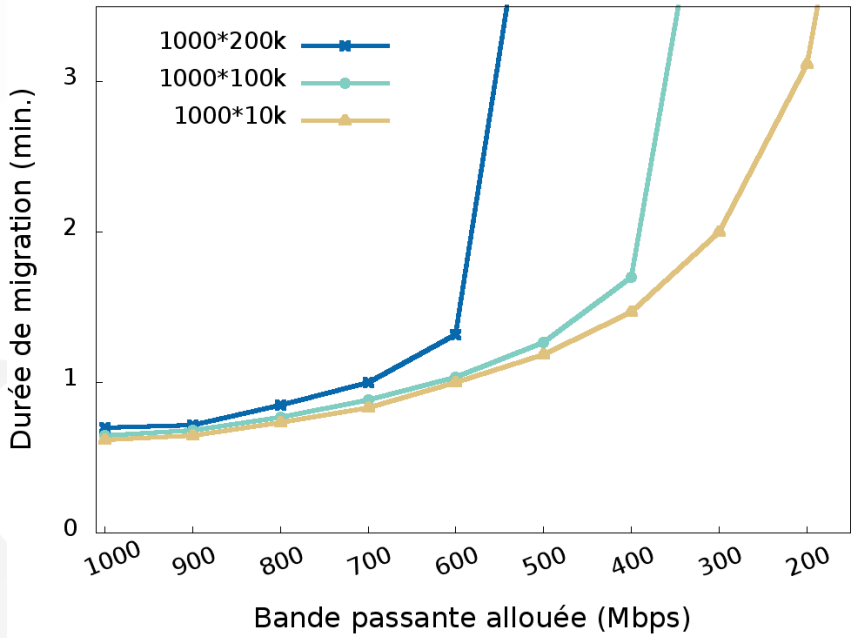
mVM : Un ordonnanceur de migrations

- Repose sur BtrPlace
  - Gestionnaire de VM
  - Placement & ordonnancement d'actions via des contraintes
  - Extensible, utilisant la programmation par contraintes
- Intègre un nouveau modèle d'ordonnancement
  - Modèle réseau
  - Modèle de migration
  - se substitue au modèle de BtrPlace
  - ~ 1600 lignes de code



Modélisation : migration à chaud

- Relation entre bande passante et durée de migration non linéaire :



- Intuition : Allouer le maximum de bande passante disponible par migration

Modélisation : estimer la durée d'une migration

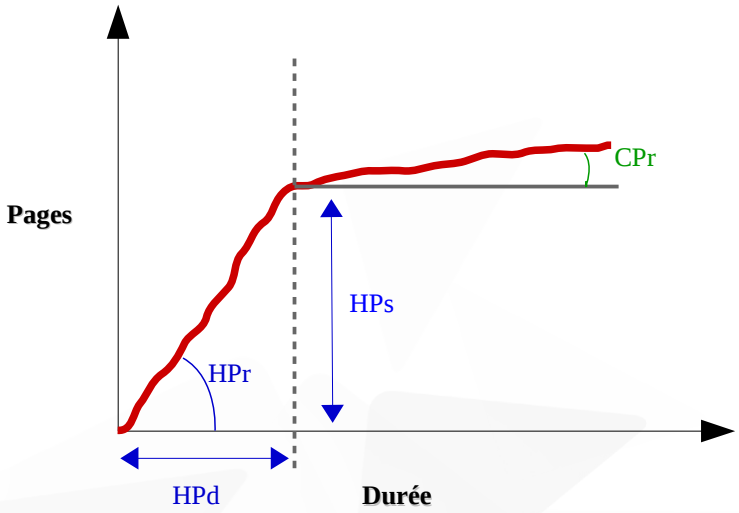
- Durée minimale (sans workload)
  - Mémoire utilisée / Bande passante [Entropy, BtrPlace, CloudSim]

- Durée effective
  - Transfert des pages mémoire réécrites

    Évolution en 2 phases :

    Hot pages → Cold pages

  - Analyse de l'activité mémoire via « libvirt »

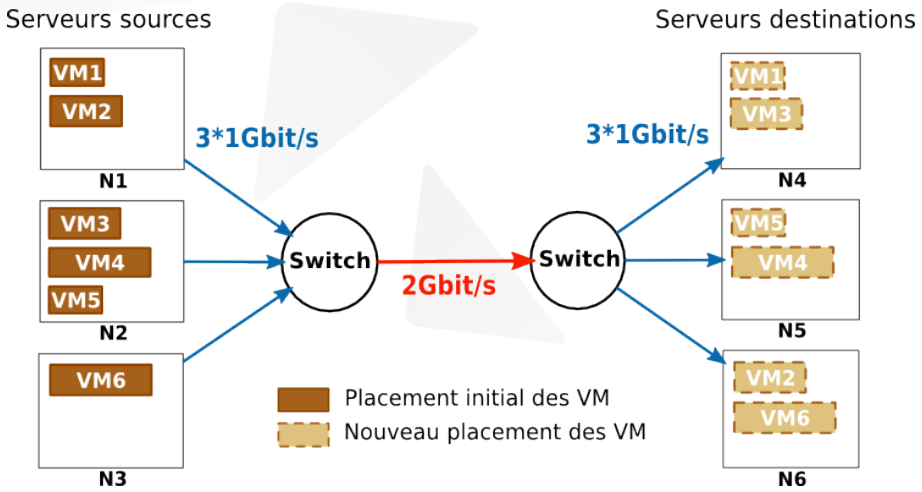


- Bande passante maximale connue
  - Pré-calcul du temps de migration

Modélisation réseau : concepts

Partage de la bande passante dans le temps

- Liens Full-duplex
- Topologies complexes
- Eléments réseaux bloquants



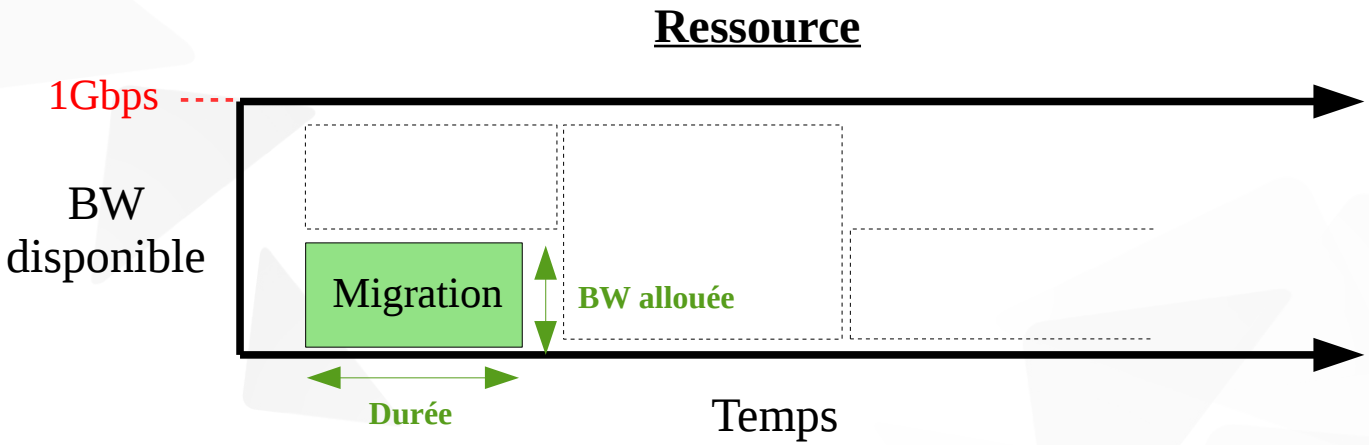
Intuitions

- Utilisation maximale de la capacité des liens
- Ne pas saturer le réseau

Modélisation réseau : implémentation

Implémentation via des contraintes « `cumulative` » :

- Placer des tâches à hauteurs et durées variables sur des ressources limitées.
- **2** ressources par lien réseau => bande passante montante et descendante
- 1 tâche <=> 1 migration



- Permet d'établir le lien entre durée de migration et bande passante à allouer

Contraintes annexes

Ajout de contraintes permettant de contrôler l'ordonnancement

- Contraintes temporelles :
  - `sync (vm[1-4]);`
  - `seq (vm[5,8]);`
  - `before (vm-1,vm-7);`
- Contrainte énergétique :
  - `powerBudget (500 Watts, [22:00-06:30]);`

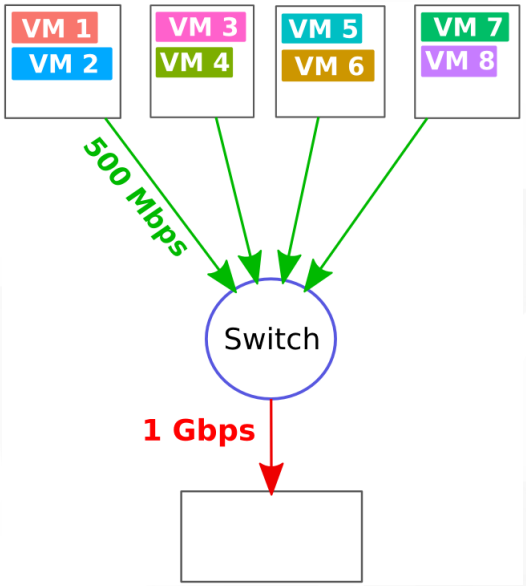
2 objectifs

- Minimiser la somme des temps de fin de migration :
  - Migrer chaque VM le plus rapidement possible
  - Assurer un faible temps de complétion
- Minimiser la consommation énergétique :
  - S'adapter à l'utilisation d'énergie renouvelable

Évaluation

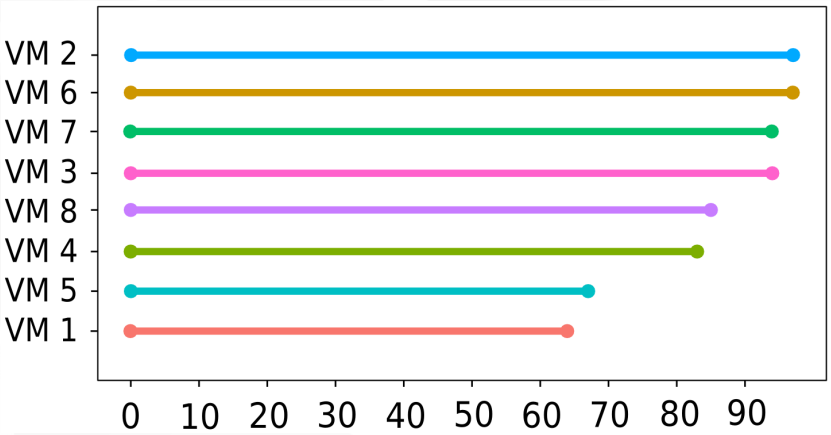
- Objectif :
  - Vérification de la précision du modèle
  - Vérification des bénéfices par rapport à BtrPlace :
    - temps individuel de migration, temps de complétion et énergie

- Configuration expérimentale :
  - Hyperviseur : KVM
  - Stockage partagé (NFS)
  - Réseau bloquant
  - Traffic shaping via la commande « tc »
  - Workloads par la commande « stress »



Évaluation

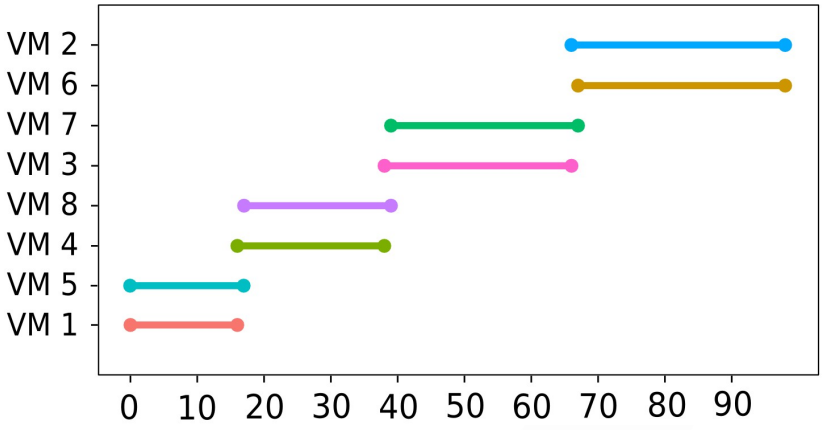
BtrPlace



- Parallélisation abusive
- Longues migrations

- Prédiction des durées **< 50%**

mVM



- Parallélisation contrôlée
- Groupement par durée

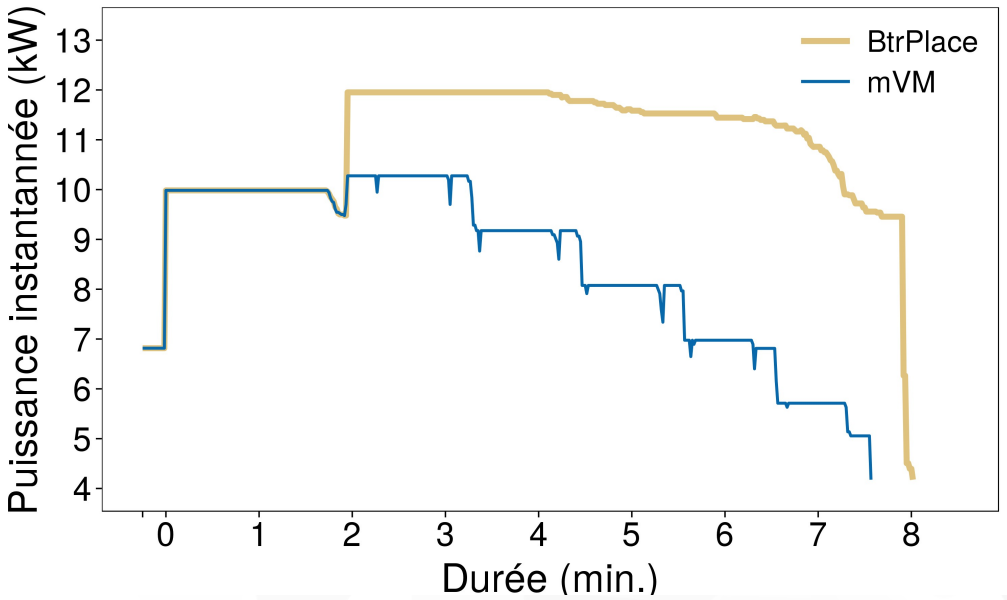
- Prédiction des durées **> 90%**
- Migrations 3.5 fois plus rapides
- Optimalité prouvée par mVM

Évaluation : énergie

- Implémentation du modèle énergétique dérivée de *[Liu et al., Cluster 2013]*
- Objectif : Minimiser l'énergie totale consommée

- Scénario de décommissionnement :
  - 48 serveurs vers 24
  - 96 VM à migrer

- Migrations 10 par 10
- Libération des noeuds au plus tôt
  - Extinction dès que possible

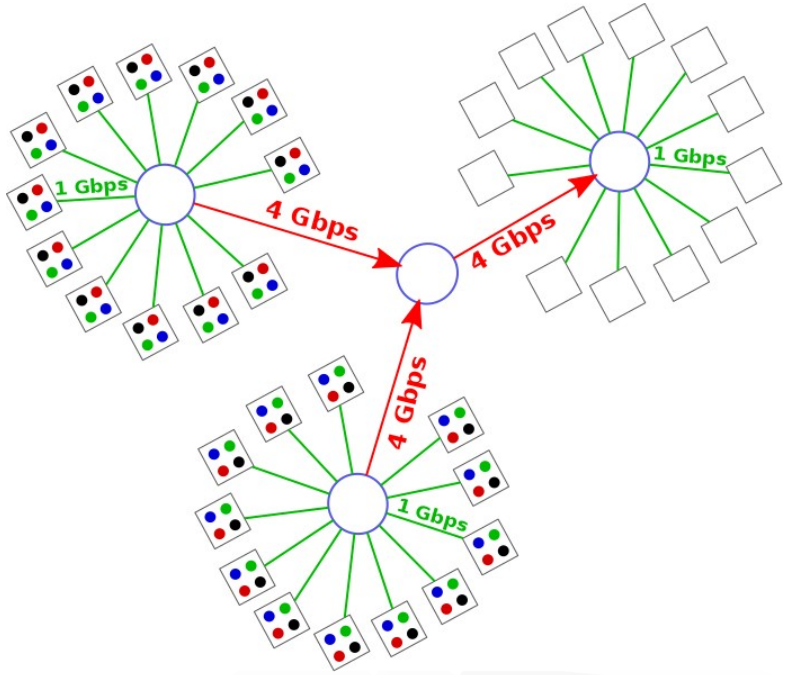


- mVM : 21.55% d'énergie sauvegardée comparé à BtrPlace

Évaluation : établir un seuil de puissance

Intérêts :

- Variabilité du coût de l'énergie
- S'adapter aux capacités de dissipation thermique
- Adaptation à la disponibilité énergétique



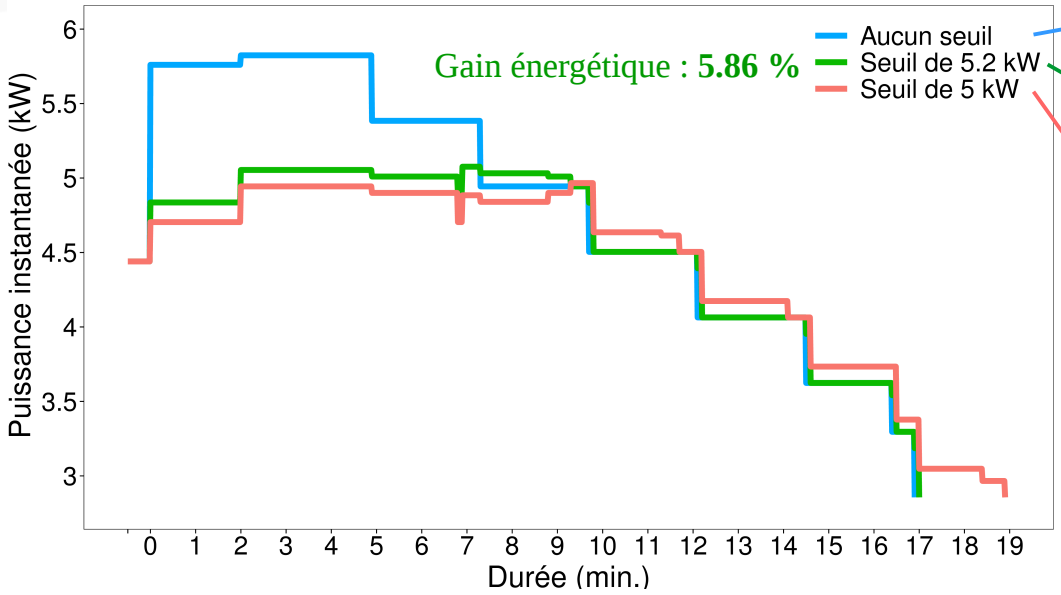
Scénario de décommissionnement :

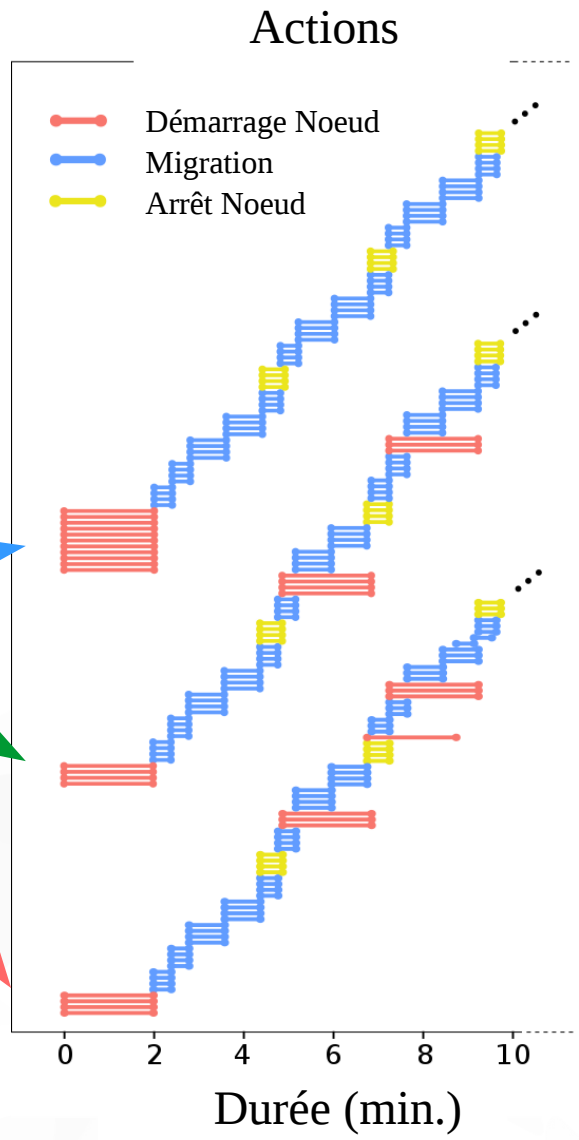
- 3 * 12 serveurs (2 racks vers 1)
- 4 VM par serveur

DC4Cities

Évaluation : contrainte 'seuil de puissance'

- Migrations 4 par 4, parallélisme optimal
  - Report des actions de boot
- 2 min. supplémentaires pour un seuil à 5kW



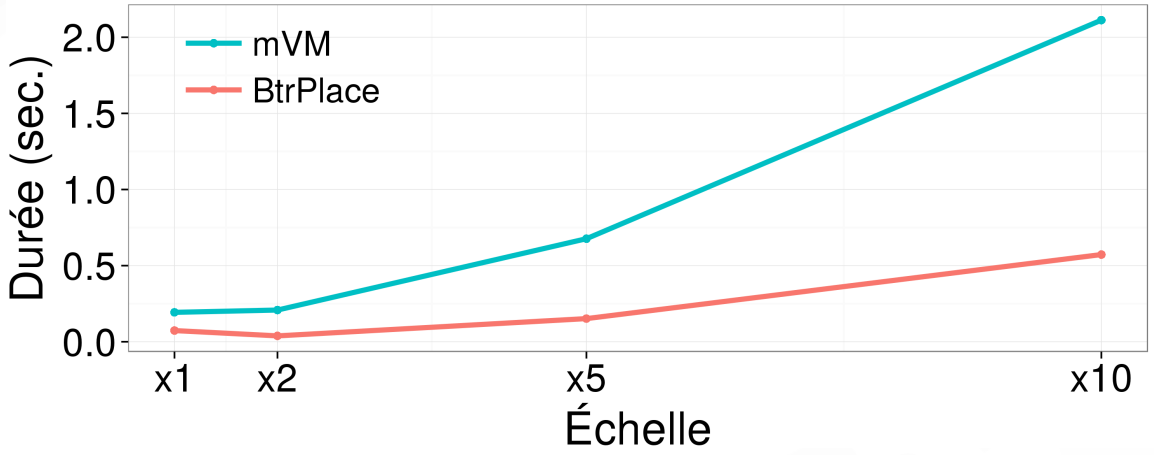


Évaluation : passage à l'échelle

Problème d'ordonnancement : NP-complet

x1
- 96 VMs
- 36 Serveurs
- Réseau : 10 Gbps

x10
- 960 VMs
- 360 Serveurs
- Réseau : 100 Gbps



1,5 secondes supplémentaires

Plus large échelle : Partitionnement des migrations.
Ex: Par cluster/rack, ..

Conclusion

Ordonnancement de migrations

- mVM considère la charge mémoire et le réseau
  - Ordonnanceur de migrations précis (> 90 %)
  - Migrations 3.5 fois plus rapides que BtrPlace
- Contrôle de l'ordonnancement via des contraintes haut niveaux
  - Synchronisation, séquentialisation / parallélisation
  - Gestion énergétique
    - - 20 % d'énergie par rapport à BtrPlace
    - Contraintes de « power capping »

Travaux futurs

- Intégration de la problématique de placement
  - Décisions de placement tenant compte de l'ordonnancement

- Downtime contrôlable => variable du modèle

Ordonnancement contrôlé de migrations à chaud

Vincent Kherbache, *Fabien Hermenier*,
Eric Madelaine