École Polytechnique de l'Université de Tours
64, Avenue Jean Portalis
37200 TOURS, FRANCE
Tél. +33 (0)2 47 36 14 14
www.polytech.univ-tours.fr

Département Informatique
5e année
2013 - 2014

Rapport PFE

Résolution d'un problème off-line de consolidation de serveurs

Encadrants
Vincent T'Kindt
vincent.tkindt@univ-tours.fr

Étudiants
Lei SHANG
lei.shang@etu.univ-tours.fr

Université François-Rabelais, Tours

DI5 2013 - 2014

Version du 20 février 2014

Table des matières

1 Cuts **5**

1.1 Cuts classiques . . . 5

1.1.1 1-Cuts . . . 5

1.2 Cuts problème-dépendent . . . 5

1.2.1 Cuts sur les contraintes des ressources . . . 5

1.2.2 Machines équivalentes . . . 6

1.2.3 Dominance des tâches* (à discuter) . . . 6

2 Test et analyse **7**

2.1 Test des 3 méthodes de résolution . . . 7

2.1.1 Méthode exacte - solveur Cplex . . . 7

2.1.2 Méthode heuristique de liste . . . 7

2.1.3 Méthode heuristique de Cplex . . . 8

2.2 Test du Preprocessing . . . 8

2.2.1 Preprocessing sans coupes supplémentaires . . . 9

2.2.2 Preprocessing avec coupe 1 . . . 9

2.2.3 Preprocessing avec coupe 2 . . . 9

2.2.4 Preprocessing avec coupe 1&2 . . . 10

2.2.5 Preprocessing avec coupe 3 . . . 10

Table des figures

Cuts

1.1 Cuts classiques

1.1.1 1-Cuts

1.2 Cuts problème-dépendent

1.2.1 Cuts sur les contraintes des ressources

Ressources CPU/GPU

Si la tâche i ne peut pas être affectée au serveur j à cause de la capacité résiduelle de CPU/GPU du serveur, alors pour toute les tâches qui demandent plus de CPU/GPU que la tâche i, cette affectation ne peut pas effectuée non plus.

Cette contrainte peut être exprimée de façon suivante :

$Si\ n^c_{i\prime} \geq n^c_i\ alors$

$$x_{i,j,t} + x_{i\prime,j,t} \leq (m^c_j - \sum_{k=1;k\neq i,i\prime;u_{k,t}=q_{k,j}=1}^{N} n^c_k x_{k,j,t})/n^c_i \qquad \forall t = 1,\ldots,T, tq\ u_{i,t} = u_{i\prime,t} = 1;$$
$$\forall j = 1,\ldots,M, tq\ q_{i,j} = q_{i\prime,j} = 1 \qquad \text{(A)}$$

$Si\ n^g_{i\prime} \geq n^g_i\ alors$

$$x_{i,j,t} + x_{i\prime,j,t} \leq (m^g_j - \sum_{k=1;k\neq i,i\prime;u_{k,t}=q_{k,j}=1}^{N} n^g_k x_{k,j,t})/n^g_i \qquad \forall t = 1,\ldots,T, tq\ u_{i,t} = u_{i\prime,t} = 1;$$
$$\forall j = 1,\ldots,M, tq\ q_{i,j} = q_{i\prime,j} = 1 \qquad \text{(B)}$$

À noter que nous n'avons pas besoin de considérer ici la contraite de préaffectation.

Ressources HDD/RAM

Les cuts sur les ressources HDD/RAM ont le même principe sauf que ces ressources puissent aussi être occupées par l'opération de la migration. Nous pouvons appliquer les mêmes cuts comme pour CPU/GPU mais la prise en compte de la migration peut rendre le cut plus strict.

$Si\ n^h_{i\prime} \geq n^h_i\ alors$

$$x_{i,j,t} + x_{i\prime,j,t} \leq (m^h_j - \sum_{k=1;k\neq i,i\prime;u_{k,t}=q_{k,j}=1}^{N} n^h_k x_{k,j,t} - \sum_{k=1;k\neq i,i\prime}^{N} \sum_{l=1;l\neq j}^{M} n^h_k y^{l,j}_{k,k,t})/n^h_i \qquad \forall t = 1,\ldots,T, tq\ u_{i,t} = u_{i\prime,t} = 1;$$
$$\forall j = 1,\ldots,M, tq\ q_{i,j} = q_{i\prime,j} = 1 \qquad \text{(C)}$$

$Si\ n^r_{i\prime} \geq n^r_i\ alors$

$$x_{i,j,t} + x_{i\prime,j,t} \leq (m^r_j - \sum_{k=1;k\neq i,i\prime;u_{k,t}=q_{k,j}=1}^{N} n^r_k x_{k,j,t} - \sum_{k=1;k\neq i,i\prime}^{N} \sum_{l=1;l\neq j}^{M} n^r_k y^{l,j}_{k,k,t})/n^r_i \qquad \forall t = 1,\dots,T, tq\ u_{i,t} = u_{i\prime,t} = 1;$$

$$\forall j = 1,\dots,M, tq\ q_{i,j} = q_{i\prime,j} = 1 \qquad \text{(D)}$$

1.2.2 Machines équivalentes

Deux serveurs j et j' peut être totalement identique dans une instance de problème si :

1. Les caractéristiques (CPU/GPU/HDD/RAM) de j et j' sont identiques
2. $q_{i,j} = q'_{i,j}$ pour $\forall i$
3. ~~j et j' ont les même voisins dans le réseau~~
4. ~~Pour chaque voisin v de j (et j'), la bande passante entre v et j est la même que celle entre v et j'~~

~~Plus largement, le j et j' ci-dessus peuvent être considérés comme un sous-réseau mais pas simplement un seul serveur. Comme ça, étant donné une solution, on peut très bien construir une autre en échangeant les affectation en j et j'.~~

[Update]Après étude du réseau, il parâit que toutes les machines sont connectées et la bande passante est unique donc on n'a besoin que de considérer les deux premières conditions.

Si j et j' sont deux machines identiques alors pour le premier instant de temps où il y a des tâches qui s'exécutent sur j ou j', alors on peut forcer que le serveur j est plus utilisé que j' pour éliminer les solutions redondantes :

$$\sum_{t_1=1}^{t-1} \sum_{i_1=1}^{N} x_{i_1,j,t_1} + ResUsed(j,t)/ResUsed(j',t) \geq 1 \qquad \forall t = 1,\dots,T;$$

$$\forall j, j' = 1 \dots M, équivalent(j,j'), j < j'; \qquad \text{(E)}$$

1.2.3 Dominance des tâches* (à voir)

Cette fois on va prendre en compte toutes les ressources requises par la tâche. On modélise 2 tâches i et $i\prime$ pour que la tâche $i\prime$ a besoin de plus de ressource que i pour tout type de ressources y compris la partie du réseau.

La formulation courante de ce cut n'est peut-être pas valide, parce que même si nous avons $x_{i,j,t} = 0$ ça ne veut pas dire qu'il n'y a pas de ressources pour la tâche i. Ça peut simplement parce que $x_{i,j,t} = 1$ ne conduit pas à la solution optimale.

Alors si nous voulons créer un Cut sur les ressources du réseau, je pense c'est quand même mieux de reprendre le même principe : si on est sûr que les ressources réseau sont insuffisantes pour la tâche i alors c'est pareil pour $i\prime$. Cependant cette modélisation (état de l'insuffisance de ressources) ne me semble pas évidente.

Test et analyse

Pour tester le fonctionnement des différentes méthodes de résolution dans ce projet, plusieurs tests ont été effectués sur un ensemble d'instance du problème. Cet ensemble contient 6 scénarios dont chacun est composé par 20 instances du problème. Toutes ces données de test sont générées par le programme Testeur.

2.1 Test des 3 méthodes de résolution

2.1.1 Méthode exacte - solveur Cplex

Dans un premier temps, le test du solveur Cplex est effectué car les solutions trouvées par le solveur peuvent nous servir à évaluer la performance des autres méthodes de résolution.

Le tableau ?? représente les statistiques effectuées sur le résultat de test du solveur Cplex. Les significations des colonnes sont :

1. Sc(N/M) : numéro de scénario et le nombre de VM et de serveur physique.
2. #Infeas : le nombre des instances qui sont prouvées comme "Infaisable" par le solveur.
3. #SolvedOpt : le nombre des instances qui sont résolues avec la solution optimale trouvée.
4. #Mem : le nombre des instances pour lesquelles le solveur n'a pas pu trouver la solution optimale à cause de la limite de l'espace mémoire.
5. #Tim : le nombre des instances pour lesquelles le solveur n'a pas pu trouver la solution optimale à cause de la limite du temps.
6. T_{min}, T_{avg}, T_{max} : le temps (minimum, moyenne et maximum) de résolution en seconde.

Sc(N/M)	#Infeas	#SolvedOpt	#Mem	#Tim	T_{min}	T_{avg}	T_{max}
Sc1(8/2)	2	18	0	0	0.02	0.07	0.15
Sc2(11/3)	9	11	0	0	0.08	0.34	1.08
Sc3(15/4)	1	19	0	0	0.59	4.33	54.57
Sc4(18/5)	0	19	0	1	1.88	239.53	1800.37
Sc5(21/5)	3	6	1	10	1.68	1196.57	1800.74
Sc6(24/6)	2	5	0	13	2.06	1316.75	1820.14

TABLE 2.1 – Résultat de test du solveur Cplex

A partir du scénario 4, on commence à avoir des grosses instances pour lesquelles le solveur n'a pas pu trouver la solution optimale ($T_{max} = 1800$) à cause de la limite de temps.

2.1.2 Méthode heuristique de liste

Le tableau ?? montre le résultat de test de la méthode heuristique de liste (on l'appelle H1 pour raison de simplicité). Dans ce tableau, les colonnes D_{min}, D_{avg} et D_{max} signifient la déviation entre la solution trouvée par H1 et la solution trouvée par le solveur Cplex. A noter que pour les 3 premiers scénarios le

solveur a trouvé la solution optimale pour toutes les instances donc cette déviation peut montrer la qualité de notre méthode H1 par rapport à la solution optimale. A partir du scénario 4 on commence à avoir des instances pour lesquelles le solveur a trouvé une solution faisable mais pas optimale à cause de la limite du temps ou de l'espace mémoire, alors dans ce cas la déviation est aussi affectée.

Sc(N/M)	#Infeas	#Solved	T_{min}	T_{avg}	T_{max}	D_{min}	D_{avg}	D_{max}
Sc1(8/2)	2	18	0.00	0.00	0.00	0%	19%	67%
Sc2(11/3)	9	11	0.00	0.00	0.00	0%	17%	46%
Sc3(15/4)	7	13	0.00	0.00	0.00	5%	24%	58%
Sc4(18/5)	11	9	0.00	0.00	0.00	10%	27%	37%
Sc5(21/5)	16	4	0.00	0.00	0.01	30%	36%	43%
Sc6(24/6)	20	0	0.00	0.01	0.02	*	*	*

TABLE 2.2 – Résultat de test de la méthode heuristique de liste (H1)

On trouve souvent 0.00 seconde dans les colonnes de temps, ce qui signifie que le temps de résolution de H1 est très court.

Par rapport à la déviation, la performance de H1 n'est pas très stable : pour certaines instances du problème, H1 a trouvé la solution optimale ($D_{min} = 0\%$) mais on a aussi dans le Sc1 $D_{max} = 67\%$ qui n'est pas très optimiste. Pour le scénario 6, H1 n'a résolu aucune instance donc la déviation n'est pas calculée.

2.1.3 Méthode heuristique de Cplex

Le tableau ?? montre le résulat de test sur la méthode heuristique basée sur le solveur Cplex (H2).

Sc(N/M)	#Infeas	#Solved	T_{min}	T_{avg}	T_{max}	D_{min}	D_{avg}	D_{max}
Sc1(8/2)	2	18	0.02	0.07	0.15	0%	0%	1%
Sc2(11/3)	9	11	0.08	0.29	0.97	0%	0%	1%
Sc3(15/4)	1	19	0.64	1.37	5.21	0%	0%	2%
Sc4(18/5)	0	20	1.53	80.33	400.32	0%	0%	2%
Sc5(21/5)	3	17	1.68	240.31	400.41	0%	2%	4%
Sc6(24/6)	2	18	2.08	286.79	410.43	-1%	1%	7%

TABLE 2.3 – Résultat de test de la méthode heuristique de Cplex (H2)

Puisque la méthode H2 est basée sur le solveur Cplex, pour les petites instances de problème elle a trouvé des solutions qui sont très proches des solutions optimales. Pour les grandes instances (à partir du scénario 4), les solutions qu'elle a trouvées sont aussi très intéressantes avec la déviation maximale égale à 7% qui reste acceptable. Le temps maximum de résolution est vers 400 secondes qui provient de l'implémentation de H2.

2.2 Test du Preprocessing

Un autre ensemble de tests a été réalisé pour évaluer la performance de l'approche Preprocessing avec des coupes différentes.

On rappelle ici l'idée du Preprocessing est de fixer autant que possible de variables dans le modèle pour réduire la taille du problème. Pour le faire, il faut avoir une borne supérieure (UB) et une borne inférieure (LB) qui sont assez proches de la solution optimale.

2.2.1 Preprocessing sans coupes supplémentaires

Nous avons lancé d'abord le Preprocessing sur toutes les variables booléennes sans ajoutant des coupes supplémentaires. Les colonnes s'interprêtent comme le suivant :

1. $DevLB_{min}$, $DevLB_{avg}$, $DevLB_{max}$: la déviation (minimum, moyenne et maximum) entre la solution optimale et LB.
2. $DevUB_{min}$, $DevUB_{avg}$, $DevUB_{max}$: la déviation (minimum, moyenne et maximum) entre UB et la solution optimale.
3. $\%Fixed_{min}$, $\%Fixed_{avg}$, $\%Fixed_{max}$: la proportion du nombre de variables fixées par rapport au nombre de toutes les variables ayant passé le preprocessing.

Dans le tableau ?? on peut trouver des statistiques sur la qualité des LB et des UB ainsi que la proportion de variables fixées pendant le Preprocessing. Nous constatons que le Preprocessing naïve ne

Sc(N/M)	LB_{min}	LB_{avg}	LB_{max}	UB_{min}	UB_{avg}	UB_{max}	$\%Fix_{min}$	$\%Fix_{avg}$	$\%Fix_{max}$
Sc1(8/2)	0.00%	4.83%	17.07%	0.00%	0.04%	0.65%	0.00%	9.25%	51.83%
Sc2(11/3)	0.04%	7.21%	17.13%	0.00%	0.01%	0.15%	0.00%	13.90%	66.67%
Sc3(15/4)	1.10%	4.01%	10.12%	0.00%	0.36%	1.80%	0.00%	0.88%	10.96%
Sc4(18/5)	0.05%	5.28%	12.56%	0.00%	0.53%	1.81%	0.00%	2.86%	55.64%
Sc5(21/5)	1.82%	5.03%	8.94%	0.56%	1.94%	7.62%	0.00%	0.00%	0.00%
Sc6(24/6)	3.94%	5.91%	8.72%	0.30%	1.75%	5.40%	0.00%	0.09%	0.56%

TABLE 2.4 – Résultat de test du Preprocessing sur toute les variables booléannes

fonctionne pas bien. Il ne fixe pas beaucoup de variables donc il n'aide pas la résolution de Cplex. Souvent, ce problème survient quand l'UB ou LB n'est pas assez bonne. En comparant la qualité de LB et de UB, on peut trouver que relativement les UB sont assez proches que la solution optimale, en revanche les LB ne sont pas très bonnes.

Pour améliorer les LB, nous avons cherché d'ajouter des contraintes supplémentaires (Cuts) au modèle LP du problème, pour enfin améliorer le fonctionnement du Preprocessing.

2.2.2 Preprocessing avec coupe 1

Nous avons d'abord relancé le test de Preprocessing avec la coupe 1 qui est problème-dépendant concernant les contraintes de ressources. Malheureusement, l'ajout de cette coupe n'a apporté aucun changement au résulat de test (identique que ??).

2.2.3 Preprocessing avec coupe 2

Cette partie contient le résultat de test de Preprocessing avec la coupe 2 (aussi notée 1-Cuts).

Dans le résultat, les trois colonnes de UB restent les mêmes car le Preprocessing n'affecte que sur les LB. Nous pouvons trouver que avec la coupe 2, le Preprocessing a pu fixé plus de variables qu'avant, surtout pour les premiers 3 scénarios. En conséquence, les LB ont été aussi améliorées (la déviation entre la solution optimale et la LB devient plus petite).

En résumé, la coupe 2 est utile pour renforcer le Preprocessing mais le résultat n'est pas encore assez bon pour bien accélérer la résolution après. A partir de scénario 4, le Preprocessing n'aide quasiment plus.

Sc(N/M)	LB_{min}	LB_{avg}	LB_{max}	UB_{min}	UB_{avg}	UB_{max}	$\%Fix_{min}$	$\%Fix_{avg}$	$\%Fix_{max}$
Sc1(8/2)	0.00 %	2.87%	17.07 %	0.00%	0.04%	0.65%	0.00%	43.91 %	100.00%
Sc2(11/3)	0.04 %	4.54%	11.97 %	0.00%	0.01%	0.15%	0.00%	15.80 %	66.67%
Sc3(15/4)	0.44 %	3.47%	9.40 %	0.00%	0.36%	1.80%	0.00%	2.22 %	10.96%
Sc4(18/5)	0.05 %	4.94%	11.60 %	0.00%	0.53%	1.81%	0.00%	2.91 %	55.64%
Sc5(21/5)	1.82 %	4.72%	8.82 %	0.56%	1.94%	7.62%	0.00%	0.00 %	0.00%
Sc6(24/6)	3.64 %	5.65%	8.45 %	0.30%	1.75%	5.40%	0.00%	0.09 %	0.56%

TABLE 2.5 – Résultat de test du Preprocessing avec la coupe 2

2.2.4 Preprocessing avec coupe 1&2

Le résultat du test de Preprocessing avec à la fois la coupe 1 et la coupe 2 reste pareil que le Preprocessing avec la coupe 2 seule. Ce fait a confirmé que la coupe 1 n'est pas valide.

2.2.5 Preprocessing avec coupe 3

Résolution d'un problème off-line de consolidation de serveurs

Département Informatique
5e année
2013 - 2014

Rapport PFE

Résumé :

Mots clefs :

Abstract:

Keywords:

Encadrants
Vincent T'Kindt
vincent.tkindt@univ-tours.fr

Étudiants
Lei SHANG
lei.shang@etu.univ-tours.fr

Université François-Rabelais, Tours

DI5 2013 - 2014