

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumedienne
Faculté d'électronique et d'informatique
Département Informatique



Mémoire du Projet de fin d'étude Pour l'obtention du diplôme d'ingénieur
d'état en Informatique

Thème

**Résolution Exacte de Problème d'Optimisation sur un
Réseau Peer-to-Peer (P2P) Basée sur le Paradigme
Master-Worker Hiérarchique**

Présenté par :

**Mr Fayçal SAADLIA
Mr Hocine MANSOURI**

Proposé et Encadré par :

Mr Ahcène BENDJOURI

Devant le Jury composé de :

**President: Mr Abdelmadjid BOUKRA
Examineur: Mr Mohamed GUERROUMI
Examinatrice : Melle Lila KHEROUA**

Promotion 2007-2008
N° d'ordre : 79 / 08

Résumé

Les problèmes d'optimisation combinatoire (*Combinatorial Optimization Problems COPS*) sont en général NP-difficiles. La résolution d'instances de grande taille de ces problèmes prend un temps de calcul exponentiel. La parallélisation du calcul est l'une des issues les plus efficaces en terme d'amélioration des performances d'exécution, notamment l'utilisation du parallélisme à grande échelle basé sur les grilles de calcul (*Grid Computing*) ou le calcul pair-à-pair (*Peer-to-Peer Computing*). Une grille de calcul, généralement utilisée pour un calcul intensif, est formée d'un nombre important de machines de calcul interconnectées par des réseaux haut débit. Le calcul pair-à-pair, est basé sur l'exploitation des cycles CPU non utilisés ou totalement oisifs sur tout le réseau. Ces deux outils sont un moyen efficace pour atteindre une grande performance de calcul pour des applications scientifiques de grande taille.

Le projet s'inscrit dans le cadre de la résolution exacte des problèmes d'optimisation de grande taille sur un système de calcul large échelle pair-à-pair. Il consiste en l'implémentation d'un nouvel algorithme parallèle basé sur un modèle Master/Worker ou Fermier/Travailleur hiérarchique avec des communications directes entre tous les processus du modèle, i.e., (Superviseur, Master et Worker). Les communications directes permettent d'une part éliminer les goulots d'étranglements qui se créent souvent au niveau des processus fermiers. Dans ces applications, les communications inter-travailleur transitent par le processus Fermier. D'autre part, les communications directes permettent de décentraliser l'espace de stockage d'unités de travail sur l'ensemble des processus Fermiers. Dans ce travail l'étudiant(e) va exploiter l'une des propriétés des réseaux P2P (communications directes entre pairs) et qui vont être mises en œuvre avec les communications de groupes en utilisant la plate-forme de calcul parallèle concurrent et distribué (ProActive).

L'algorithme implémenté sera appliqué au problème du *flow-shop de permutation (PFS)*. L'application sera implémentée sur la plateforme pair-à-pair *ProActive*.

Mots-clés : Optimisation combinatoire exacte, GRID Computing, P2P Computing, ProActive, PFS, QAP, Modèle Master/Worker.

Table des matières

Glossaire	i
Table des matières	ii
Liste des figures & tableaux	vi
Introduction Générale	1
1 Théorie de la Complexité & Méthodes d'Optimisation	
Combinatoire	4
1.1 Introduction.....	4
1.2 Théorie de la complexité.....	5
1.2.1 Complexité des algorithmes.....	5
1.2.2 Ordre de grandeurs.....	6
1.2.3 Complexité des problèmes.....	6
1.2.3.1 Classification des complexités.....	7
1.2.3.1.1 Les problèmes de classe P.....	7
1.2.3.1.2 Les problèmes de classe NP.....	8
1.2.3.1.3 Les problèmes NP-complets.....	8
1.2.3.1.4 Les problèmes NP-Difficiles.....	8
1.3 Approches de Résolution de problèmes d'optimisation.....	9
1.3.1 Introduction.....	9
1.3.2 Les Méthodes Exactes.....	9
1.3.2.1 Le Branch & Bound (Séparation et Evaluation progressive)....	10
a) La technique de Séparation	10
b) La méthode d'évaluation.....	10
c) La méthode de sondage.....	10
d) La méthode de sélection.....	10
1.3.2.2 La programmation dynamique.....	11
1.3.2.3 La programmation linéaire.....	12
1.3.2.4 Méthode d'énumération A [*]	12
1.4 Les Méthodes Approchées (Heuristiques & Métaheuristiques).....	13
1.4.1 Heuristiques.....	13
1.4.2 Métaheuristiques.....	13
• Les algorithmes génétiques.....	13

• colonies de fourmis.....	14
1.5 Conclusion.....	14
2 Les Réseaux Pair-à-Pair, les grilles de calcul et le P2P-Computing..	15
2.1 Introduction.....	15
2.2 Grid Computing (Grille de Calcul).....	16
2.2.1 Définition.....	16
2.2.2 Architecture de grille de calcul (Grid).....	17
2.2.3 Les caractéristiques des grilles de calcul.....	18
2.2.3.1 Existence de plusieurs domaines administratifs.....	18
2.2.3.2 Hétérogénéité des ressources.....	18
2.2.3.3 Passage à l'échelle (scalability).....	18
2.2.3.4 Nature dynamique des ressources.....	18
2.3 Les Réseaux P2P.....	18
2.3.1 Définitions.....	18
2.3.2 Caractéristiques des réseaux P2P.....	21
2.3.2.1 Architectures P2P.....	21
2.3.2.1.a) Architecture centralisée vers décentralisée.....	22
2.3.2.1.b) Architecture décentralisée (pur p2p).....	22
2.3.2.1.c) Architecture Hybride (modèle Super Nœuds).....	23
2.3.2.2 Dynamisme.....	24
2.3.2.3 Hétérogénéité.....	24
2.3.2.4 Extensibilité.....	24
2.3.2.5 Disponibilité.....	25
2.3.2.6 Tolérance aux pannes.....	25
2.3.2.7 Sécurité.....	25
2.3.3 Domaines d'application des technologies P2P.....	25
2.3.3.1 Partage de fichier.....	25
2.3.3.2 Les plates formes p2p.....	26
2.3.3.3 Le calcul distribué.....	26
2.3.4 Quelques systèmes pair -à- pair et de grille.....	26
2.3.4.1 Napster-Gnutella.....	26
2.3.4.2 Kazaa.....	28
2.3.4.3 Globus.....	28
2.3.4.4 GRIDRPC.....	29

2.3.4.5	Condor.....	29
2.3.4.6	Seti@Home.....	30
2.3.4.7	Grid'5000.....	31
2.3.4.8	Boinc.....	31
2.3.4.9	XtremWeb.....	31
2.3.4.10	JXTA.....	32
2.3.4.11	ProActive.....	34
3	ProActive, une Plate-forme Pair-à-Pair.....	35
3.1	Introduction.....	35
3.2	Modèle de programmation avec objets actifs.....	36
3.2.1	Concept d'objet Actif et d'objet Passif.....	36
3.2.1.1	Création d'un objet actif.....	39
3.2.1.1.a)	Création à l'instanciation.....	39
3.2.1.1.b)	Création à partir d'un objet existant.....	40
3.2.1.1.c)	Spécification des <i>nœuds</i> d'accueil des objets actifs... ..	40
3.2.1.1.d)	Enregistrement des objets actifs.....	40
3.2.1.2	Concept de groupe d'objets.....	41
3.3	Les groupes de communication.....	42
3.4	Mécanisme de déploiement.....	42
3.5	La Structure Pair à Pair de ProActive.....	44
3.6	Fonctionnement de la plateforme.....	45
3.6.1	Le Bootstrapping.....	45
3.6.2	La dynamique et l'auto organisation.....	46
3.6.3	Les différents paramètres de l'infrastructure.....	47
3.7	Conclusion.....	47
4	Les Algorithmes Branch and Bound Parallèles	48
4.1	Introduction.....	48
4.2	Le Branch and Bound.....	48
4.2.1	Algorithme Branch and Bound.....	49
4.2.1.1	Illustration de la méthode sur un exemple de problème.....	50
4.2.1.2	Algorithme Branch and Bound Séquentiel.....	52
a)	Le Branchement.....	53

b) Le Bounding.....	53
c) L'Elimination.....	53
d) La Sélection.....	54
4.2.1.3 Les connaissances générées par un B&B.....	54
4.3 Algorithme Branch & Bound Parallèle.....	55
4.3.1 Un Modèle d'Algorithme B&B Parallèle.....	55
(A1) Filiation directe.....	55
(A2) Synchronicité	55
(A3) Granularité constante.....	56
(A4) Aucun surcoût d'implémentation.....	56
(A5) La communication.....	56
4.3.2 Les Architectures Parallèles.....	56
4.3.3 Propriété caractéristique.....	57
4.3.3.1 B&B Séquentiel Optimal.....	57
Définition 3.1.....	57
4.3.3.2 Notion de Consistance.....	57
Définition 3.2.....	57
4.4 Types de Parallélisation d'un Branch and Bound.....	58
4.4.1 Parallélisme de type 1 (<i>ou noeud-basé</i>)	58
4.4.2 Parallélisme de type 2 (<i>ou arbre basé</i>).....	58
4.4.3 Parallélisme de type 3 (<i>ou multi-recherche</i>).....	59
4.4.4 Hybridation entre les trois types.....	59
4.5 Facteurs d'implémentation des algorithmes à Arbre basé (type 2).....	59
4.5.1 Le synchronisme.....	59
4.5.2 Les espace de travail.....	60
4.6 Performance de B&B Parallèle.....	61
Définition 4.6	61
Définition 4.7	61
4.7 Vers un Modèle Asynchrone.....	61
Définition 4.8.....	61
4.8 Discussion.....	62
4.9 Conclusion.....	63
5 Le Paradigme Master-Worker et le Master-Worker Hiérarchique..	64
5.1 Introduction.....	64

5.2	Les contraintes liées au parallélisme.....	65
5.2.1	Distribution.....	65
5.2.2	Communication.....	65
5.2.3	Synchronisation.....	66
5.3	Paradigme master-worker.....	66
5.3.1	Définition.....	66
5.3.2	Cas d'utilisation.....	67
5.3.3	Quelques travaux master/worker effectués.....	67
5.3.3.1	Linderoth et Yoder (2000).....	68
5.3.3.2	Aida, Natsume et Futakata (2002).....	68
5.3.3.3	Mezmaz et al (2007).....	69
5.3.3.4	Bendjoudi, Melab et Talbi(2007).....	69
5.4	Les inconvénients d'un master-worker.....	70
5.5	Paradigme Master-Worker Hiérarchique.....	71
5.5.1	Exemples des travaux master-worker hiérarchique.....	71
5.5.1.1	Aida et Al (2003-2004).....	71
5.5.1.1.1	Le processus Superviseur.....	73
5.5.1.1.2	Le processus Master.....	73
5.5.1.1.3	Le processus Worker.....	74
5.5.1.1.4	Implémentation sur Grille de Calcul.....	75
5.5.1.2	Alexandre Di Costanzo (2007).....	76
5.5.1.2.1	Le processus Master.....	77
5.5.1.2.2	Le processus Sub-master.....	77
5.5.1.2.3	Le processus Worker.....	77
5.5.1.2.4	Le processus Leader.....	77
5.6	Conclusion.....	78
6	Conception	80
6.1	Introduction.....	80
6.2	Problématique.....	81
6.3	Donnée du Problème.....	81
6.4	L'algorithme Branch and Bound Parallèle proposé.....	82
6.4.1	L'opération du Branchement.....	83
6.4.2	L'opération de la Sélection.....	85
6.4.3	L'Elimination.....	86

6.4.4	Distribution des sous-problèmes sur les masters.....	86
6.4.5	Distribution des tâches sur les workers.....	87
6.4.6	Communication et échange de connaissances entre les différents Composants.	88
6.4.6.1	Communications Superviseur/Master.....	90
6.4.6.1.1	Le Superviseur vers un Master.....	90
6.4.6.1.2	Un Master vers le Superviseur.....	90
6.4.6.2	Communications Inter-Master.....	90
6.4.6.3	Communications Master/Worker.....	91
6.4.6.3.1	Un Master vers un Worker	91
6.4.6.3.2	Un Worker vers un Master.....	91
6.4.6.4	Communications Worker/Worker.....	91
6.4.6.4.1	Inter-worker	91
6.4.6.4.2	Intra-worker.....	92
6.4.6.5	Communications Worker/Superviseur.....	92
6.5	Redistribution en cas de pannes ou de perte.....	92
6.6	Les Algorithmes.....	93
6.6.1	Algorithme du Superviseur.....	93
6.6.2	Algorithme du Master.....	94
6.6.3	Algorithme du Worker.....	95
6.7	Application à un Flow Shop de permutation.....	96
6.7.1	Notations.....	97
6.7.2	La borne inférieure pour le Makespan utilisée	98
6.7.2.1	Le Flow Shop à deux machines $F2 \text{permut} C_{\text{Max}}$	98
6.7.2.2	La borne inférieure pour Le Flow Shop à m machine.....	99
6.8	Conclusion.....	100
7	Implémentation & Tests sur ProActive	101
7.1	Introduction.....	101
7.2	Environnement de Mise en Œuvre.....	102
7.3	Le Schéma Fonctionnel de l'application.....	102
7.3.1	Niveau 0.....	102
7.3.2	Niveau 1.....	102
7.3.3	Niveau 2.....	102
7.3.3	Niveau 3.....	103

7.4	Fonctionnement de HParallelBB sur un réseau P2P.....	103
7.4.1	Phase d'initialisation et de déploiement.....	103
7.4.2	Phase de distribution des sous-problèmes sur les Master.....	105
7.4.4	Le partage de connaissances et la communication entre les différentes entités.....	107
7.4.5	Prise en charge des nouvelles arrivées (pairs).....	108
7.5	Tests et résultats.....	110
7.5.1	Tests sur Grille.....	110
7.5.1.1	Présentation de GRID5000.....	111
7.5.1.2	OAR.....	111
7.5.2	Tests en mode Pair-à-Pair.....	112
7.6	Conclusion.....	114
	Conclusion Générale et Perspectives.....	115
	Bibliographie.....	116