


République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari BOUMEDIENE

Faculté d'Electronique et d'Informatique
Département Informatique



Mémoire du projet de fin d'études
Pour l'obtention du diplôme
D'Ingénieur d'Etat en Informatique

THEME

Les algorithmes de colonies de fourmis parallèles
 pour la résolution des problèmes d'optimisation
 combinatoire sur grille

Réalisé par:

KHAIDA Takioueddine Hamza
 BOUALLOUCHE Hamid

Thème proposé par:

M^r BENDJOURI Achène

Soutenu le: 07/10/2008

Devant le jury composé de :

M ^r	BOUKRA Abdelmadjid	: Président
M ^r	GHEROUMI Mohammed	: Examineur
M ^{me}	KHEROUA Lila	: Examineur

PROMOTION : 2007 / 2008



Remerciements

Louange à Dieu le tout puissant qui nous a guidés pour l'accomplissement de ce travail, et on ne saura nous faire guider sans le guide de Dieu, Dieu merci.

On tient tous d'abord, à exprimer notre profonde reconnaissance et notre gratitude à notre promoteur Mr BENDJOUDI qui nous a proposé le sujet et à toute l'équipe CERIT, et à notre Co-promoteur Mr BOUKRA de nous avoir permis de s'investir dans un domaine aussi intéressant que les algorithmes de colonies de fourmis, pour nous avoir soutenus, dirigés et orientés durant toute l'année par leurs précieux conseils, et qui ont été d'un soutien moral et surtout pour leurs disponibilités qu'on nous n'oublierons jamais. Pour toutes ces raisons on tient à réitérer le témoignage de notre profonde gratitude.

On remercie une autre fois Mr BOUKRA pour l'honneur et le plaisir qu'il nous a fait en présidant ce jury.

On remercie également Mr GUERROUMI et Mlle KHEROUA pour avoir accepté de juger notre travail.

Nos profonds remerciements aux enseignants de l'USTHB pour leurs enseignements, soutiens et conseils qu'ils nous ont promulgués.

Enfin, nous remercions toutes les personnes qui nous ont aidées de près ou de loin.

Merci.

Dédicaces

Je dédie ce travail :

A mes très chers Parents, les êtres les plus chers à mon cœur sans qui je ne serais arrivé où j'en suis aujourd'hui et auxquels je témoigne ma plus profonde reconnaissance pour le sacrifice et les encouragements constants tout au long de mes études.

A ma grande mère, que dieu la protège.

A mes frères : Ahmed, Saleh Eddine.

A mes sœurs : Fatima Zohra, Fouzia, Samra, KHaïra, Hanane, Fadhila, Nawel et la petite belle fille Chaimaa.

A toute ma Famille: mes oncles, mes tantes, mes cousins et cousines, paternels qu'ils soient ou maternels.

A mon binôme KHAÏDA Hamza, pour son sérieux et son soutien moral.

A mes meilleurs amis: Amine, Anis, Bilal, Zinou, Mourad, Mahdi, Abdeslam, Lyes, Abdelmalek, Hamid, Arezki et Samir..., pour leurs soutiens et avec qui j'ai partagé mes plus belles années, spécialement à mes meilleurs amis Abdelkader Et Mohamed qui ont toujours été présents dans mes joies et mes peines et que j'espère garder éternellement.

Et que tous ceux que je n'aurais pas cités m'excusent.

A tous ceux qui ne pourraient assister à ma consécration.

Je le dédie enfin à tous ceux qui croient que ce travail mérite d'être poursuivi.

Et en particulier à tous le monde.

Abdelhamid BOUALLOUCHE

Résumé

La majorité des problèmes d'optimisation combinatoire sont NP-difficiles. Tenter de les résoudre avec une méthode exacte s'est avéré irréalisable du fait du temps de calcul excessif. Des méthodes approchées ont été développées pour une résolution rapide de ces problèmes, mais force est de constater que le temps de calcul que mettent les méthodes approchées est considéré comme raisonnable par rapport au temps de calcul excessifs des méthodes exactes. La parallélisation des méthodes approchées fournit des algorithmes efficaces pour la résolution des problèmes d'optimisation combinatoire, l'utilisation du déploiement à large échelle sur des systèmes distribués Pair à Pair (P2P), basé sur l'exploitation des cycles CPU non utilisés, offre un moyen efficace pour atteindre une grande performance de calcul par une distribution des calculs. Dans notre travail, nous nous intéressons à la résolution approchée d'un problème d'ordonnancement de type FLOWSHOP de permutation mono critère (Makespan), qui est un problème largement étudié dans la littérature. Nous avons proposé une parallélisation de l'algorithme de colonie de fourmis, que nous l'avons appliqué à ce type de problèmes.

Mots Clés : *Optimisation combinatoire, Flowshop de permutation, Colonies de Fourmis, Colonies de Fourmis Parallèle, Calcul PeertoPeer, ProActive.*

Glossaire

P2P	Peer to Peer : Un système pair-à-pair
AO	Active Object : Un Objet Actif
AN	Active Node : Un Noeud Actif
ACO	Ant Colony Optimisation : Un algorithme approché pour la résolution des problèmes l'optimisation combinatoire.
AS	Ant System : Premier algorithme proposé pour la résolution de TSP
JVM	Java Virtual Machine : Machine Virtuelle Java.
TSP	Traveling Salesperson Problem : problème de voyageur de commerce
ACS	Ant Colony System : L'amélioration de AS.
ProActive	Une plateforme pair-à-pair dédiée pour le calcul parallèle, distribué et concurrent.
ParallelACO	Parallel Ant Colony Optimisation : La parallélisation de ACO proposée dans notre travail.

Table des matières

Glossaire	i
Table des matières.....	ii
Liste des figures.....	vii
Liste des tableaux	viii
Introduction Générale.....	1

Chapitre I : Problèmes d'optimisation combinatoire et méthodes de résolution

1. Introduction	4
2. Théorie de complexité	4
3. Classes de complexité	4
3.1 Classe P	5
3.2 Classe NP	5
3.3 Classe NP-Complet	5
4. Problème de voyageur de commerce	6
4.1 Complexité	7
4.2 Intérêt	7
5. Problème d'ordonnancement de la production	8
5.1 Définition	8
5.2 Les concepts de base	8
5.2.1 Les tâches	8
5.2.2 Les Ressources	8
5.2.3 Les critères d'optimisation	9
5.2.4 Les contraintes	9
5.3 L'environnement machine	9
5.3.1 Les problèmes à une machine	9
5.3.2 Les problèmes d'atelier multi-machines	9
6. Problèmes de décision et problèmes d'optimisation	10
7. Problèmes d'optimisation combinatoire	11
8. Approches de résolution	11
8.1 Les méthodes exactes	11
8.1.1 La programmation linéaire	12
8.1.2 La programmation dynamique	12
8.1.3 La méthode Branch and Bounds	12
8.2 Les méta-heuristiques	14
8.2.1 Les méta-heuristiques à solution unique	15
8.2.1.1 Les méthodes de descente	15
8.2.1.2 Le recuit simulé	15
8.2.1.3 La recherche Tabou	16
8.2.2 Les méta-heuristiques à population de solutions	17
8.2.2.1 Les algorithmes génétiques	17
8.2.2.2 Les colonies de fourmis	19
9. Conclusion	19

Chapitre II : Les réseaux Pair à Pair et Les grilles de calcul

1. Introduction	21
2. Les grilles de calcul	21
2.1 Historique et définitions	21
2.2 Définitions	22
2.3 Architecture générale d'une grille	23
3. Réseaux pair à pair (P2P)	24
3.1 Présentation	24
3.2 Définitions	25
3.3 Classification des systèmes informatiques	25
3.4 Objectifs des systèmes P2P	26
3.5 Caractéristiques des systèmes P2P	26
3.6 P2P contre Client/Server	27
3.7 Les architectures des réseaux P2P	28
3.7.1 Les architectures centralisées	28
3.7.2 Architectures décentralisées	29
3.7.3 Architecture hybride	30
3.7.3.1 les super-peer	30
3.7.3.2 Les super-peer redondants	31
4. Domaines d'application des technologies P2P	32
4.1 Calcul distribué	32
4.2 Diffusion des données	32
4.3 Le travail collaboratif	33
4.4 Les Plate-formes P2P	33
5. Quelques exemples d'applications des technologies P2P et de Grille	34
5.1 GNUTELLA	34
5.2 Freenet	35
5.3 KaZaA	35
5.4 Skype	36
5.5 EDonkey	36
5.6 JXTA	37
5.7 XtremWeb	37
5.8 Grid'5000 exemple de grille	37
5.9 SETI@Home	38
5.10 ProActive	38
6. Conclusion	39

Chapitre III : Les algorithmes de colonies de fourmis séquentiels et parallèles

1. Introduction	41
2. Historique	41
3. Optimisation par colonie de Fourmis	42
3.1 Le comportement des fourmis réelles	42
3.2 Les fourmis artificielles	43
4. La Métaheuristique	44
4.1 Ant System	44
4.2 Ant Colony System	46
5. Aspects liés à la parallélisation	47
5.1 Les architectures parallèles	48
5.2 La synchronisation	49

5.3	La communication	49
6.	Parallélisation de l'OCF	49
6.1	Classification des Algorithmes Parallèle de l'OCF	50
6.1.1	Classification selon le nombre de fourmis	50
6.1.1.1	Distribution d'une Simple Colonie (fine- granularité)	50
6.1.1.2	Distribution à multi Colonies (gros grain)	50
6.1.2	Classification selon le mode de communication	53
6.2	Topologies d'intercommunication	54
6.2.1	Topologie Tout à Tout	54
6.2.2	Remplacer-le plus mauvais	54
6.2.3	Hypercube	54
6.2.4	Topologie Anneau	54
6.3	Modèles parallèle de l'OCF	54
6.3.1	Modèle synchrone master/workers	54
6.3.2	Modèle Asynchrone	55
6.3.3	Le modèle de diffusion	56
6.3.4	Modèle à passage de messages vs. Modèle à mémoire partagée	56
6.4	Stratégies d'échange d'information	58
6.4.1	Échange de la meilleure solution globale	58
6.4.2	Échange circulaire des meilleures solutions locales	58
6.4.3	Échange circulaire de migrants	59
6.4.4	Échange circulaire des meilleures solutions locales et des migrants	59
7.	Conclusion	59

Chapitre IV : La plate-forme ProActive

1.	Introduction	61
2.	Concept d'objet Actif et d'objet Futur	61
3.	les composants d'un objet actif	62
3.1	Le rôle du Stub	63
3.2	Le rôle de Proxy	63
3.3	Le rôle du Corps (Body)	63
3.4	Le rôle de l'instance standard de la classe Worker	64
4.	Les appels asynchrones et les objets futurs	64
4.1	Les objets futurs	64
4.2	Les appels asynchrones	65
5.	Création d'un objet actif	67
5.1	Création à l'instanciation	67
5.2	Création à partir d'un objet existant	68
5.3	Spécification des noeuds d'accueil des objets actifs	68
5.4	Enregistrement des objets actifs	68
6.	Concept de groupe d'objets	69
7.	Mécanisme de déploiement	69
8.	Les agents mobiles et de la migration	71
9.	L'API Master/Workers	72
9.1	Aperçu	72
9.2	Usage et principe	72
9.2.1	Le déploiement du Framework	73
9.2.2	La définition et l'affectation des tâches	73
9.2.3	La récolte des résultats	73

9.3	L'API en détail	74
9.3.1	La création et le déploiement du master	74
9.3.1.1	La création d'un master local	75
9.3.1.2	La création d'un master distant	75
9.3.1.3	L'ajout des ressources (workers)	75
9.3.2	La définition des tâches	76
9.3.3.1	Principe	76
9.3.3.2	Usage	77
9.3.3	La mémoire d'un worker	76
9.3.4	L'affectation des tâches	77
9.3.5	La récolte des résultats	78
9.3.6	La période des ping	78
9.3.7	La terminaison du master	78
10.	Conclusion	79

Chapitre V : Conception

1.	Introduction	81
2.	Le problème du flowshop de permutation	81
3.1	Les données de production	81
3.2	Fonction objectif	82
3.3	Modélisation du problème	83
3.4	Représentation d'un ordonnancement	85
4.	L'algorithme d'ACO parallèle proposé	86
4.1	Communications et échange d'informations	87
4.1.1	Le master vers le worker	87
4.1.2	Un worker vers le master	87
4.1.3	Un worker vers un worker	87
4.2	Tolérance en cas de panne	87
5.	Les algorithmes	87
5.1	Algorithme du Master	88
5.2	Algorithme du Worker	90
6.	Conclusion	91

Chapitre VI : Implémentation et tests

1.	Introduction	94
2.	Environnement de mise en oeuvre	94
3.	Phase de distribution du calcul sur les workers	95
4.	Les communications entre les différentes entités	96
5.	Prise en charge des nouveaux arrivés (pairs)	98
6.	Tolérance aux pannes	99
7.	Fonctionnement de ParallelACO	100
8.	Tests	104
8.1	GRID'5000	104
8.2	Prise en main de l'environnement	104
8.2.1	Première connexion	104
8.2.2	Configurer pour réserver	105
8.2.3	Superviser et réserver avec OAR	105

8.2.3.1 Oarsub	105
8.2.3.2 Le mode interactif	105
8.2.3.3 Le mode passif	106
8.3 Tests en mode peer-to-peer	106
9. Conclusion	109
Conclusion Générale	110
Bibliographie.....	112