Université Cadi Ayyad ENSA Marrakech Génie Informatique

Année universitaire 2017-2018

# RAPPORT DE PROJET DE SEMESTRE

3ème Année Génie Informatique

**Application d’une heuristique sur un problème de placement dans un environnement distribué.**

Effectué à l’ENSA de Marrakech

Réalisé par Encadré par

|  |  |
| --- | --- |
| DAOULAT NIZAR  DEHBAOUI DOUNIA  [LAHIAOUNI ABDERRAHMAN](https://plus.google.com/u/0/107396681553359890251?prsrc=4) | Mr. ELMARZOUQI  Mme.ZRIKEM  Mme.AITAMHIRA |

Devant le jury

MR. ELMARZOUQI PROFESSEUR A L’ENSA DE MARRAKECH

MME. ZRIKEM PROFESSEUR A L’ENSA DE MARAAKECH

MME. AITAMHIRA DOCTORANTE A L’ENSA DE MARAAKECH

Soutenu le 16 janvier 2018

Remerciement

*Nous aimerions consacrer cette section pour remercier particulièrement notre chers professeurs et encadrants, Mr. ELMAROUQI, Mme ZRIKEM et Mme AITAMHIRA d’une part pour avoir choisi un sujet très riche et constructif, et d’autre part pour tous ses bons conseils qui nous ont permis de mener à bien notre projet.*

Table des matières

[RAPPORT DE PROJET DE SEMESTRE 1](#_Toc503810046)

[**Chapitre 1 : Mise en situation** 3](#_Toc503810047)

[I. Introduction 3](#_Toc503810048)

[II. Définition système distribué 3](#_Toc503810049)

[III. Objectif général 4](#_Toc503810050)

[VI. Conclusion 4](#_Toc503810051)

[**Chapitre 2 : Problématique** 5](#_Toc503810052)

# **Chapitre 1 : Mise en situation**

## Introduction

Vue l'augmentation de leurs ressources, les entreprises multi-sites doivent pouvoir compter sur une infrastructure informatique hautes performances, à même d'assurer l'exécution transparente de leurs processus informatiques et une communication fiable, en interne (entre les différents sites) comme en externe (avec les partenaires et clients). Cela exige une surveillance continue de la disponibilité des ressources et de l'utilisation de la bande passante des réseaux localement distribués.

## Définition système distribué

Un système distribué est un système disposant d'un ensemble d'entités communicantes, installées sur une architecture d'ordinateurs indépendants reliés par un réseau de communication, dans le but de résoudre en coopération une fonctionnalité applicative commune.

Autrement dit, un système distribué est défini comme étant un ensemble des ressources physiques et logiques géographiquement dispersées et reliées par un réseau de communication dans le but de réaliser une tâche commune. Cet ensemble donne aux utilisateurs une vue unique des données du point de vue logique.

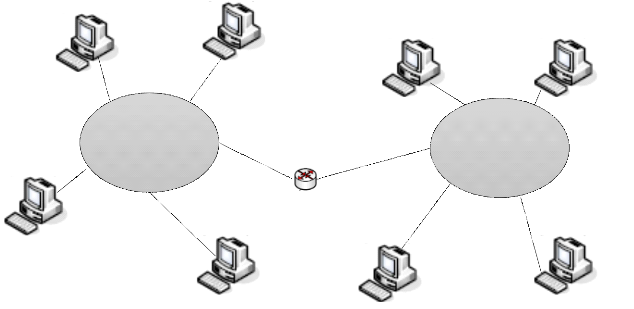
Un système distribué est un ensemble d'entités autonomes de calcul (ordinateurs, PDA, processeurs, processus, processus léger etc.) interconnectées et qui peuvent communiquer.

Figure1 : représentation d’un système distribué

## Objectif général

L’objectif de notre projet faire l’étude et la résolution d’un problème de déploiement dans un système distribue telle que on a un ensemble de processus qu’il faut dépoiler dans différentes serveuses tout en respectant les contraintes et de trouver une meilleure solution en terme du temps en vérifiant un critère particulier défini par une fonction objective.

## VI. Conclusion

Au cours de ce chapitre, nous avons défini le système distribue et l’objectif principale de notre projet, le chapitre suivant détaillera notre problématique.

# **Chapitre 2 : Problématique**

## Introduction

Ce chapitre décrit le problématique de déploiement sur un environnement répartie et ses difficultés.

## Problématique

Le déploiement des composants devient de plus en plus complexe du fait de la diversité des terminaux d’accès et des infrastructures de communication. La plupart des outils de déploiement existants utilisent des techniques qui ne sont pas à la mesure de la complexité des problèmes rencontrés, tels que :

– L’hétérogénéité des machines destinations.

– Les conflits de déploiement entre les composants.

– Les exigences requises des composants.

– Le choix de placement des composants sur les nœuds de déploiement.

– La nature et la topologie des réseaux sur lequel s’effectuera le déploiement.

Le placement de composants sur les nœuds répartis est devenu une étape cruciale dans le déploiement complet d’une application sur les machines destinations.

En effet, le choix du placement influence énormément les performances de l’application, ainsi que le rendement et la consommation des différentes ressources utilisées par l’application.

Le déploiement doit prendre en compte les besoins, les conditions et les contraintes fonctionnelles de l’application à déployer. Ainsi, que les propriétés non fonctionnelles et la disponibilité des ressources qui sont souvent partagées avec des accès réglementés. Il faut aussi respecter les exigences du réseau sur lequel on déploie, par exemple les ressource de serveur à ne pas dépasser. Le déploiement doit aussi assurer un meilleur rendement et des performances optimales du système global qui sera déployer.

* **Les difficultés du déploiement répartis :**

Lorsqu’il s’agit d’un déploiement réparti sur plusieurs machines, beaucoup de difficultés qui n’existent pas dans le cas du déploiement monoposte surgissent.

On peut les résumer comme suit :

– Le choix du placement des processus sur les différents nœuds.

– La connaissance de la topologie du réseau et les nœuds qui le composent.

– L’hétérogénéité des machines et des réseaux sur lesquelles on réalise le déploiement.

– La communication entre les différents processus de l’application à déployer.

## Conclusion

Après avoir le problématique de notre projet nous allons passer à la formulation du problème.

# **Chapitre 3 : Formulation**

Ce qu’on a :

1. Un ensemble de nœuds/sites P= {p1, p2, …, pn}
   1. Chaque pi peut effectuer une quantité de travail en une unité de temps : Puissance du site Ppi
   2. Chaque pi a une quantité de ressource Rpi.
   3. Chaque pi peut communiquer avec tout pk avec une vitesse de transmission de données de : Nik.
2. Un ensemble de processus T= {t1, t2, …, tm}
   1. Chaque ti a une quantité de travail Tti.
   2. Chaque ti consomme une quantité de ressource Rti.
   3. Chaque tj communique une quantité de donne avec un tk de Qjk

On note :

-L’ensemble des processus déployés sur le site pi Di

-Une solution est l’ensemble des Di avec i, et Di et Dj sont distinctes.

Ce qui est demande :

1. Déployer l’ensemble des processus T sur l’ensemble des nœuds P, en tel sort que :
   1. Minimiser les couts de communication.
   2. Minimiser les couts d’exécution.

Contraintes :

* Pour chaque Di : Rpi ≥ j , avec tj

Fonction objectif :

1. Cout d’exécution :
   1. Cout d’exécution d’un site i : CEpi= Ppi . , avec tj
   2. Cout d’exécution totale :

CE= Max (CEPi) avec

1. Cout de Communication :
   1. Cout de communication entre tous les processus hébergés sur le même site : Par convention ça sera égale à zéro.
   2. Cout Totale de communication :

Avec tj et tk

Finalement le cout total est de Ctotale=CE+CC

# **Chapitre 5 : Solution**

# Introduction

Dans ce chapitre nous allons détailler la solution adoptée pour la réalisation de ce projet.

## Métaheuristique

### 5.2.1. Définition

Une métaheuristique est un [algorithme](https://fr.wikipedia.org/wiki/Algorithme) d’[optimisation](https://fr.wikipedia.org/wiki/Optimisation_(math%C3%A9matiques)) visant à résoudre des problèmes d’[optimisation difficile](https://fr.wikipedia.org/wiki/Th%C3%A9orie_de_la_complexit%C3%A9_des_algorithmes) (souvent issus des domaines de la [recherche opérationnelle](https://fr.wikipedia.org/wiki/Recherche_op%C3%A9rationnelle), de l'[ingénierie](https://fr.wikipedia.org/wiki/Ing%C3%A9nierie) ou de l'[intelligence artificielle](https://fr.wikipedia.org/wiki/Intelligence_artificielle)) pour lesquels on ne connaît pas de méthode classique plus efficace.

Les métaheuristiques sont généralement des algorithmes [stochastiques](https://fr.wikipedia.org/wiki/Stochastique) itératifs, qui progressent vers un optimum global, c'est-à-dire [l'extremum global d'une fonction](https://fr.wikipedia.org/wiki/Extremum#Extrema_d'une_fonction), par [échantillonnage](https://fr.wikipedia.org/wiki/%C3%89chantillonnage) d’une [fonction objectif](https://fr.wikipedia.org/wiki/Optimisation_(math%C3%A9matiques)). Elles se comportent comme des algorithmes de recherche, tentant d’apprendre les caractéristiques d’un problème afin d’en trouver une approximation de la meilleure solution (d'une manière proche des [algorithmes d'approximation](https://fr.wikipedia.org/wiki/Algorithme_d%27approximation)).

Il existe un grand nombre de métaheuristiques différentes, allant de la simple [recherche locale](https://fr.wikipedia.org/wiki/Recherche_locale_(optimisation)) à des algorithmes complexes de recherche globale. Ces méthodes utilisent cependant un haut niveau d’abstraction, leur permettant d’être adaptées à une large gamme de problèmes différents.

### 

### 5.2.2. Tabou

La recherche tabou est une [métaheuristique](https://fr.wikipedia.org/wiki/M%C3%A9taheuristique) d'optimisation présentée par [Fred W. Glover](https://fr.wikipedia.org/w/index.php?title=Fred_W._Glover&action=edit&redlink=1) [(en)](https://en.wikipedia.org/wiki/Fred_W._Glover) en 1986. C’est une méthode heuristique de recherche locale utilisée pour résoudre des problèmes complexes et/ou de très grande taille (souvent NP-durs).

* Principe de base :

Poursuivre la recherche de solutions même lorsqu’un optimum local est rencontré et ce,

■ en permettant des déplacements qui n’améliorent pas la solution

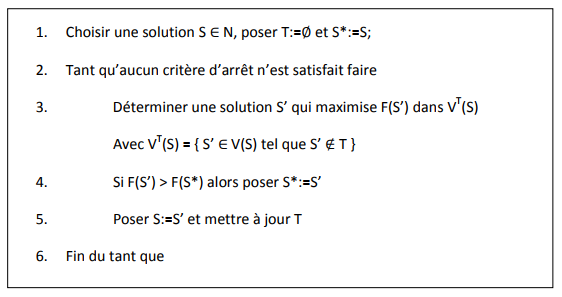
■ en utilisant le principe de mémoire pour éviter les retours en arrière (mouvements cycliques)

* Algorithme de la recherche Tabou

Soit :

S : Solution courante ;

S\* : Meilleure solution rencontrée depuis le début de la recherche



* Algorithme adapter pour notre problème :

A l’état initiale on génére une collection aléatoire de , Telle que on déployer aléatoire en ensemble processus sur les nœuds et on insère notre solution initiale sur tabou.

Ensuite, On cherche un voisinage de la solution meilleur et qui n’appartient pas à la liste tabou, meilleur .

*Le mouvement primitif :*

*On prend un processus Pi d’un serveur et on l’héberge sur un serveur et on interdire le mouvement inverse*

Après, si cette solution vérifier la contrainte et meilleur que la solution initiale alors on garde et mise à jour la liste tabou.

### 5.2.3. Colonie d’abeille

* Comportement d’une colonie d’abeille :

Une colonie d’abeille cherche à produire le maximum de miel à partir du nectar, dans un minimum de temps, pour cela chaque abeille fait un tour sur un champ pour récolter le maximum de nectar à partir de différente fleurs, sans dépasser la capacité de son réservoir. Chaque fleur visitée consomme une durée de temps, et charge les pattes de l’abeille avec une quantité de pollen.

Une abeille termine son voyage si son réservoir de nectar est plein ou bien s’il ne reste plus de fleurs à visiter, et à la fin de voyage elle ramène le nectar récolté au ruche et décrit son voyage ou autre abeilles, cette historique est réservé au mémoire des abeilles de ruche pour optimiser les futurs voyages.

* Similarités entre le problème de déploiement et la colonie d’abeille :

On peut voir le problème comme: un ensemble de nœuds qui héberges différentes combinaisons de processus, et terminer avec un cout d’exécution et de communication minimales et fortes compatibilité entre les processus d’un même nœud.

D’autre part une colonie d’abeille : c’est un ensemble d’abeilles qui traitent -extraction de nectar- un champ de fleurs et le transformer en miel.

|  |  |  |
| --- | --- | --- |
|  | Colonie d’abeille | Problème de déploiement |
| Traiteur | Abeille | Nœud/site/serveur |
| Traité | Fleur | Processus |
| Nature de traitement | Extraction du nectar | Exécution |
| Cout traitement | Temps de traitement | Cout d’exécution |
| Puissance de traitement | Qté Travail /unité de temps | Qté Travail /unité de temps |
| Entre les traités | Miscibilité entre les nectars issues de différentes fleures | Compatibilité entre les Processus dans un même nœud |
| Charge de traitement | Une fleur charge une abeille avec une qté de pollen | Un processus consomme une qté de ressource d’un nœud |
| Entre les traiteurs | Cout de miscibilité entre les miels produit par différents abeilles | Cout de communication entre différents nœuds |
| Conditions de non traitement | Incompatibilité du nectar de la nouvelle fleur avec un autre déjà existant dans le réservoir | Incompatibilité du nouveau processus avec un autre déjà hébergé |
| Insuffisance d’espace pour le pollen sur l’abeille | Insuffisance de ressources sur un nœud |
| Toutes les fleurs sont traitée | Tous les processus sont déjà hébergé. |
| Objectifs | Durée de voyage courte | Cout d’exécution minimale |
| cout de miscibilité | Cout de communication minimale |
|  |  |

|  |  |
| --- | --- |
| Abeille artificielle | Abeille naturelle |

* Chaque abeille a un cout pour adapter son miel avec les autres abeilles.
* Une abeille naturelle ne manque pas d’espace pour le pollen, en fait il charge et décharge d’une fleur a l’autre.
* Une abeille naturelle peut visiter et retraiter une fleur déjà visitée, par contre une abeille artificielle ne vas pas visiter une fleur déjà traitée par elle ou par une autre abeille.
* Reformulation du problème de déploiement :

Ce qu’on a :

1. Un ensemble de d’abeilles P= {p1, p2, …, pn}
   1. Chaque pi peut effectuer une quantité de travail en une unité de temps : Puissance de l’abeille Ppi
   2. Chaque pi a un volume de réservoir de pollen Rpi.
   3. Chaque pi peut mélanger son miel avec tout pk avec une vitesse de miscibilité de : Nik.
2. Un ensemble de fleurs T= {t1, t2, …, tm}
   1. Chaque ti a une quantité de travail Tti.
   2. Chaque ti consomme un volume du réservoir de pollen Rti.
   3. Chaque tj nécessite une quantité de travail pour que son nectar se mélange avec celle d’une fleur tk de Qjk

On note :

-L’ensemble des fleurs traite par l’abeille pi Di

-Une solution est l’ensemble des Di avec i, et Di et Dj sont distinctes.

Ce qui est demande :

1. Traiter un ensemble de fleur T (extraction de nectar) par un ensemble d’abeille P, en tel sort que :
   1. Minimiser les couts de mélange.
   2. Minimiser les couts de traitement.

Contraintes :

* Pour chaque Di : Rpi ≥ j , avec tj

Fonction objective :

1. Cout de traitement :
   1. Cout de traitement d’une fleur i : CEpi= Ppi . , avec tj
   2. Cout de traitement totale :

CE= Max (CEPi) avec

1. Cout de mélange :
   1. Cout de mélange entre tous les nectars extractés par la même abeille : Par convention ça sera égale à zéro.
   2. Cout Totale de mélange :

Avec tj et tk

Finalement le cout total est de Ctotale=CE+CC

* Fonctionnement :

La colonie se compose de plusieurs troupes NT (nombre de troupes dans la colonie)

P11 P12 … P1n

P21 P22 … P2n

… … … …

PNT1 PNT2 … PNTn

Chaque ligne représente une troupe (notre set de nœuds).

Chaque colonne représente duplication d’une abeille (même caractéristiques)

Chaque colonne d’abeille partage une mémoire commune sur les fleurs de champs.

Initial :

Toutes les fleurs sont marquées comme non visitée.

Pas de mémoire sur les fleurs

Pour chaque abeille de la troupe devant une fleur, il peut qu’elle la traite ou pas d’une probabilité de 50%. On arrête lorsque toutes les fleurs sont traitées

A la fin du voyage du troupe chaque abeille à réaliser le traitement a une durée de temps (cout de traitement) et produit son propre miel

Donc l’abeille s’engage à mélanger les différents nectars qu’elle a collecter, mais pour les mélanger avec celles d’autre abeille chaque nectar de fleur i impose un cout pour être mélanger avec le nectar d’une autre fleur j (cout de communication).

Cout Totale de mélange des nectars :

Avec tj et tk

Durée totale de voyage CE= Max (CEPi) avec

C’est la durée entre le temps de commencement du voyage jusqu’à l’arrivée de de la dernière abeille de la troupe.

Le cout total est de Ctotal= CE + CC

A la fin d’un voyage d’une abeille il partage son expérience avec les abeilles d’autres troupes de sa colonne, pour cela il renforce la mémoire sur des nœuds avec :

=

Quantité de mémoire ajouter pour une fleur tj par l’abeille pi dans la classe k (colonne)

Q = constante

Alpha et beta sont des variables de contrôle.

t : exprime le numéro d’itération.

(Le choix se pose : chaque troupe peut met à jour la mémoire sur le champ dés quelle termine son voyage ou bien on attend, la fin des voyages de toutes les troupes pour mettre à jours les mémoires pour l’itération suivante)

Dans un premier lieu on choisit la 2eme possibilité. La comparaison entre les deux se fait à base des statistiques d’expériences.

Le terme de mémorisation devient :

A la fin de toutes les voyages de toutes les troupes un facteur d’oubli s’introduit pour éviter les minimas locaux et permet à visiter des nouveaux types de solutions.

Donc la mise à jour des mémoires devient :

 : Coefficient d’oubli, définie la vitesse de perte de mémoire des abeilles.

Choix d’une fleur :

Une abeille pi pendant un voyage, et pour le choix de fleur à traiter, c’est une décision aléatoire de 50%.

Si l’abeille décide de traiter une fleur alors le choix de la fleur et aussi aléatoire avec une probabilité de choisir la fleur tj de:

Rpi(t) : espace dans le réservoir de pollen.

Rtj: quantité de pollen que la fleur tj vas déposer.

N(t) : ensemble des fleurs non traiter a l’instant t.

N.B : if ne faut pas initialiser les mémoires par 0, ça engendre des problèmes, par contre il faut l’initialiser avec une constante très faible.

# Conclusion

Ce projet est une occasion qui nous a permis de bien saisir l’importance sur le système distribue et les métaheuristique. Ainsi, nous avons touché de près les difficultés relatives à la réalisation d’un projet de ce genre. Il a été tellement fructueux dans la mesure où il nous a offert la possibilité de concrétiser ce que nous avons étudié. Aussi, en développant ce projet, un grand travail de recherche a été entrepris.