

Approche distribuée de découverte de services dans Globus Toolkit 4

Marie Hélène Wassa Mballo

Département d'informatique
Université Cheikh Anta Diop
SENEGAL

mballomaria@yahoo.fr

Ibrahima Niang

Département d'informatique
Université Cheikh Anta Diop
SENEGAL

ibrahima_niang@hotmail.com

Yahya Slimani

Département d'informatique
Université El Manar Tunis
TUNISIE

yahya.slimani@fst.rnu.tn

RÉSUMÉ. Les grilles informatiques sont des systèmes distribués à large échelle dont l'objectif est l'agrégation et le partage de ressources hétérogènes géographiquement réparties pour répondre aux besoins des applications de haute performance. La découverte de services constitue une problématique de recherche majeure surtout dans les grilles. Cependant, les approches proposées présentent des limites pour la recherche de services à large échelle, car ces elles se focalisent plus sur l'estimation des ressources matérielles de calcul. Dans cet article, nous proposons une méthode distribuée de recherche de services plus optimisée basée sur une représentation des services sous forme d'arbre.

ABSTRACT. Grid computing are distributed and scalable systems whose objective is the aggregation and the sharing of heterogeneous resources for high performance applications needs. Service discovery is one of the major problems for these grids. However, the proposed approaches have limitations for search services in large scale, because they are more focused on estimating resources computing. In this paper, we propose an optimized distributed method of services discovery with services representation based on a tree.

MOTS-CLÉS : grille de calcul, découverte de services, P2P.

KEYWORDS: grid computing, service discovery, P2P.

1. Introduction

Les grilles informatiques sont perçues aujourd'hui comme des fournisseurs de services. Cependant, un des problèmes essentiels relatif à l'exploitation de ces grilles est la découverte de services [1] [2]. En effet, les services peuvent être définis comme un ensemble d'applicatifs que des serveurs mettent à disposition des utilisateurs. Durant ces dernières années, nous avons assisté à une intégration des services web et des grilles. Cette combinaison permet de résoudre le problème d'hétérogénéité des applications. Dans cette nouvelle architecture, l'UDDI (Universal Description Discovery and Integration Framework) joue un rôle très important car il constitue l'annuaire qui permet la publication et l'accès aux services. C'est pourquoi le système de grille Globus, dans sa dernière version, a migré entièrement vers les services web basé sur un composant nommé Service Index similaire au fonctionnement de l'UDDI. Cependant, cette dernière version de Globus présente une architecture hiérarchique qui a des limites dans le cadre de découverte de services à large échelle. Notons que la plupart des solutions proposées sont basées sur des architectures P2P (Peer to Peer) hybrides ou structurées. Ces approches reposent surtout sur des algorithmes de recherche qui s'adaptent seulement sur la découverte des configurations matérielles des machines de la grille.

Dans cet article nous proposons une méthode de découverte de services distribuée basée sur une architecture hybride qui permet une recherche efficace à large échelle avec l'utilisation d'un arbre des services optimisant le temps de recherche dans un pair donné de la grille. Le reste de l'article est structuré de la manière suivante : dans la section 2, nous présentons le background de notre travail. Les approches de découverte de services sont proposées dans la section 3. Dans la section 4, nous présentons une approche de découverte de services distribuée. Une analyse théorique est proposée dans la section 5. Enfin, nous terminons par une conclusion et des perspectives dans la section 6.

2. Background

Notons que la découverte de ressources dans les grilles est une tâche complexe vue que la grille est composée d'un nombre important de ressources, répartis à travers plusieurs domaines administratifs géographiquement distribués. Les solutions de découverte de ressources récentes dans Globus [3] intègrent les services web afin d'offrir une interopérabilité entre les environnements hétérogènes. Pour cela, un composant similaire à l'annuaire UDDI, nommé Index Service [4], permet de stocker et de découvrir les services publiés dans la grille. Cependant, Globus présente un système hiérarchique qui n'est pas adapté dans un environnement dynamique et à large échelle.

3. Approches de découverte de ressources dans les grilles

3.1. Solutions distribuées

Dans [5], il a été proposé un système distribué des annuaires UDDI basé sur un réseau structuré (DHT). Le réseau DHT agit comme un réseau de rendezvous permettant la connexion de plusieurs annuaires UDDI pour promouvoir la scalabilité des services d'information dispersés à travers plusieurs annuaires. L'idée principale de ce travail est la gestion des services à large échelle. Il résout le problème de scalabilité des services web. Concernant le processus de découverte, les critères de recherche sont hachés pour obtenir la clé associée aux ressources recherchées dans le réseau DHT. Cependant, vu que la recherche se fait sur la base de clé, le nombre de résultats obtenu est très limité

[6] propose une approche P2P hybride où les ressources de calcul de la grille sont divisées en trois classes : **machine unique** qui est l'unité de base, **cluster homogène** composé d'un ensemble de pairs ayant les mêmes propriétés et le **cluster hétérogène** composé d'un ensemble de pairs dont les caractéristiques sont différentes. Dans cette approche, pour un sous réseau donné les pairs sont organisés de manière hiérarchique et le super pair est chargé d'établir la communication avec d'autres sous réseaux. Pour la découverte de ressources, deux acteurs sont mis en jeu : **user agent (UA)** et le **discovery agent (DA)**. L'UA transfère une requête au DA qui est exécuté dans le super pair S. le DA sera chargé de transférer la requête à ses DA voisins de réseau. Cette approche présente cependant des limites car elle est efficace si la recherche se limite juste aux caractéristiques des ressources. Mais dans le cas de recherche de services, il y aura des contraintes vu l'organisation hiérarchique des pairs qui peut entraîner un temps de recherche très long.

3.2. Problématiques de découverte de ressources dans les grilles

Globus Toolkit 4 [7] présente une architecture hiérarchique basée sur un Service Index qui centralise les données des autres Service Index de la grille. Notons que dans la découverte de services, nous devons tenir compte de la dynamique du système et la dispersion géographiques des ressources. Or la plupart des approches proposées se focalisent surtout sur la découverte des caractéristiques matérielles. En plus, les solutions basées sur une architecture structurée présente des limites dans la découverte de services, car elles sont basées sur l'utilisation de clé, ce qui réduit considérablement le nombre de résultats fournis.

4. Une approche de découverte de services distribuée à large échelle

4.1. Présentation et Architecture

4.1.1. Présentation

Notre solution s'appuie sur une structuration des services sous forme d'arbre et une plate-forme de recherche basée sur une architecture P2P hybride. Chaque pair stocke des services qui sont représentés par un arbre. En effet, l'arbre permet une meilleure représentation des données et permettant un accès rapide en réduisant la masse d'informations à traiter. Concernant la représentation des services, nous avons choisi des paramètres permettant de les caractériser :

- Nom : chaque service de la grille a un nom permettant de l'identifier dans une organisation virtuelle (VO) [8] donnée.
- Licence : ce paramètre permet de savoir si le service est libre ou propriétaire.
- Catégorie : permet de savoir à quelle catégorie le service appartient.
- Adresse : ce paramètre définit l'adresse du pair hébergeant le service.

4.1.2. Architecture

Comme dans le Globus Toolkit 4, notre système de grille est composé d'une collection de VOs et dont chacune d'elle est indexée par un Service Index. La figure 1 montre notre architecture de recherche proposée qui s'appuie sur deux composants : la **couche P2P** et le **Client appli**. La grille est divisée en plusieurs VOs, chaque VO contient des **pairs (P)** et un **super pairs (SP)** qui permet la communication entre les VOs. Le super pair est choisi selon sa puissance de calcul et sa capacité de stockage. Le fonctionnement de cette architecture est la suivante. La **couche P2P** contient l'ensemble des super pairs de chaque VO de la grille. Pour joindre le réseau P2P, le super pair doit juste connaître au moins l'URL d'un super pair. Chaque pair stocke les services qu'il fournit, et dispose d'un annuaire de services. Lorsqu'un nouveau service doit être ajouté, tous les paramètres de recherche sont fournis tels que : le nom, la licence, la catégorie et l'adresse. Ces paramètres sont enregistrés dans l'arbre de services. Le super pair est également mis-à-jour avec des nouveaux services ajoutés dans une VO. Lors de la recherche de services, une requête est envoyée en local sur un pair, avec un ou plusieurs paramètres de recherche définis précédemment. Si aucun résultat n'est fourni pour la recherche en local, il est possible de transférer la requête aux autres pairs de la VO par l'intermédiaire du super pair. Ce dernier consulte son arbre pour savoir les pairs susceptibles de fournir les résultats attendus. Le super pair est également chargé de transférer la requête à ses super pairs voisins. Ces derniers à leur tour transfèrent la

requête. Le **Client Appli** initialise la requête en local. Si aucun résultat n'est fourni, la requête est transférée au super pair, qui est chargé de faire suivre la requête aux autres pairs ou supers pairs.

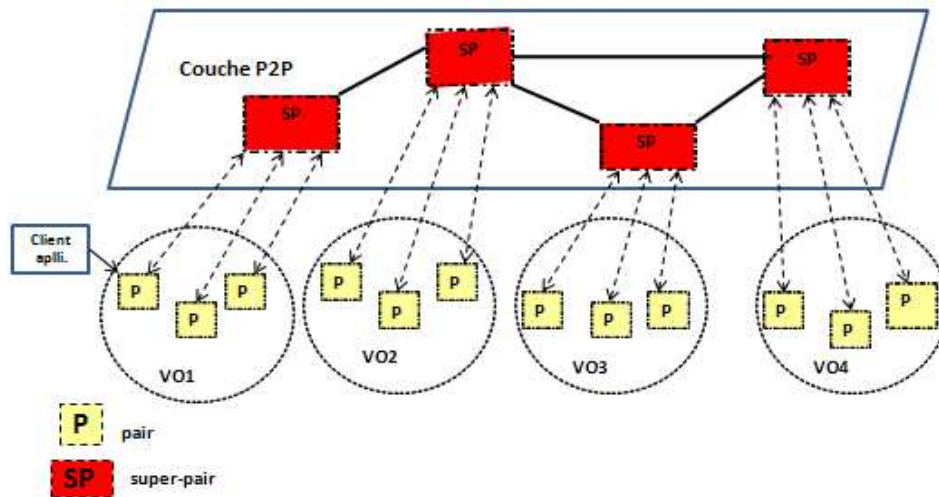


Figure 1. Architecture de recherche

4.2. Structuration des services

Contrairement aux services Index de Globus qui s'adaptent mieux à la recherche de composants matériels, nous proposons une structuration sous forme d'arbre qui permet d'une part d'optimiser les temps de recherche mais d'autre part d'avoir une meilleure représentation des services. Ainsi, notre arbre a une hauteur fixe égale à cinq (5) avec les niveaux suivants: niveau 0 : contient le nom de l'organisation virtuelle où sont publiés les services ; niveau 1 : a comme valeur le type de licence du service. Sur ce niveau nous ne pouvons avoir au maximum que deux valeurs (libre ou propriétaire), niveau 2 : contient la catégorie des services. Ce paramètre permet de savoir le type du service ; niveau 3 : contient l'adresse des machines où les services sont publiés ; niveau 4 : contient le nom des services mises en œuvre dans la grille.

La figure 2 montre un arbre de services d'une VO donnée. Chaque pair contient un arbre de services. Puisque notre arbre a une hauteur fixe, la croissance se fait en largeur. Par contre le niveau 1 a une largeur fixe, puisqu'au maximum il ne peut avoir que deux valeurs possibles. Ainsi, la croissance de l'arbre de services dépend des niveaux deux (2), trois (3) et quatre (4).

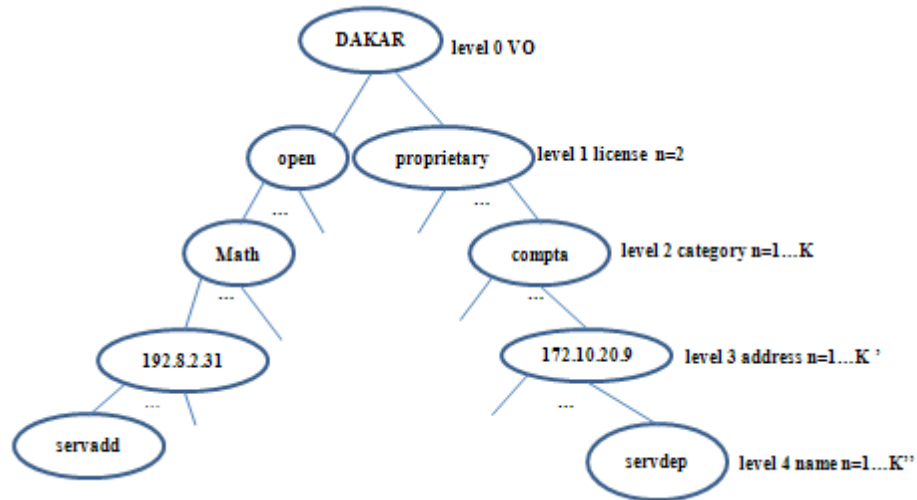


Figure 2. Arbre des services

4.3. Mécanisme de découverte des services

Nous distinguons deux types de recherche : la recherche locale et la recherche globale : i) **Découverte locale** : le Client Appli soumet la requête à un pair, en donnant les paramètres de recherche. Si la recherche en local ne fournit aucun résultat, la requête est transmise au super pair de la VO. Le super pair pourra contacter des pairs susceptibles de contenir le (les) service (s) demandé (s), une fois son arbre consulté. Un certain temps est attendu, si aucun résultat n'est fourni, la recherche s'élargit à large échelle. ii) **Découverte globale** : la découverte locale s'est avérée infructueuse. Dans ce cas le super pair transfère la requête par inondation à ses voisins proches. Ces derniers reprennent le même processus de recherche. Cependant le nombre de sauts de la requête est limité par le TTL (Time-To-Live).

5. Discussion

5.1. Analyse de l'arbre de services

Pour une meilleure représentation des services et permettre d'avoir un temps de recherche efficace, nous utilisons un arbre. La complexité de notre arbre va dépendre des niveaux deux(2), trois (3) et quatre (4), donc notre arbre grandit en largeur. Il permet de réduire la masse d'informations à traiter car une recherche dichotomique est effectuée de la gauche vers la droite. Nous allons calculer la complexité de notre arbre en considérant les paramètres suivants :

x = nombre maximum de nœuds logiques du niveau 2 ;

y = nombre maximal de nœuds fils du niveau 2 ;

z = nombre maximal de nœuds fils du niveau 3 ;

Compte tenu de la définition de la complexité d'un algorithme [9]. L'opération utilisée dans la recherche est une comparaison ce qui veut dire que la comparaison se fait avec les valeurs stockées au niveau de chaque nœud. De ce fait notre complexité au pire des cas est égal à :

- (1) $Ct = O(x+y+z)$; dans le cas, où un arbre de recherche n'est pas utilisé, la complexité dans le pire des cas serait de :
- (2) $Ct = O(4xyz)$; 4 dans la formule (2) représente les quatre niveaux de un (1) à quatre (4) de l'arbre de services.

La comparaison du (1) et du (2) montre réellement que l'arbre permet un accès plus rapide à l'information. Donc théoriquement l'approche de l'arbre est meilleure.

5.2. Analyse de l'architecture

La technologie P2P permet une découverte à large échelle, en effet la découverte globale est gérée par la couche P2P contenant l'ensemble des super pairs des différentes VO de la grille. Avec l'approche P2P hybride, si la recherche au niveau local n'aboutit pas il directement transféré au super pair. Dans la recherche, le nombre de message émis que ça soit en local ou en global est réduit. En effet en local si au niveau du pair aucun résultat n'est fourni, il est directement routé au super pair, et au niveau global le routage de la requête est limité par un TTL. Cependant, avec la structure hiérarchique, un seul pair centralise l'ensemble des données des autres pairs de la grille. Il ya une répétition des données dans les différents pairs. Cette approche a tendance à augmenter le temps de recherche si nous devons effectuer une découverte à grande échelle.

6. Conclusion et perspectives

Dans cet article, nous nous sommes intéressés au problème de découverte de services à large échelle dans les grilles informatiques. Nous avons vu que les solutions distribuées proposées utilisent la technologie P2P qui se focalisent sur la découverte des configurations matérielles. Nous avons proposé une approche de découverte adaptée pour les services qui sont l'ensemble des applicatifs mis à la disposition des utilisateurs. Notre solution permet une découverte à large échelle utilisant une plate-forme basée sur le P2P hybride avec un temps de recherche optimisé grâce à l'arbre de recherche utilisé. Comme perspectives, nous envisageons de passer à l'étude expérimentale de notre système de découverte.

6. Bibliographie et Biographie

6.1. Bibliographie

- [1] by Steven Fitzgerald , Ian Foster , Carl Kesselman , Gregor von Laszewski , Warren Smith Steven Tuecke, A Directory Service for Configuring High-Performance Distributed Computations, In *proc. 6th ieee symp. on high performance distributed computing*, 1997.
- [2] Luis Ferreira,Viktors Berstis, Jonathan Armstrong, Mike Kendzierski, Andreas Neukoetter, MasanobuTakagi, et al., *Introduction to Grid Computing with Globus*, Copyright IBM Corp. 2003.
- [3] I. Foster, C. Kesselman, Globus: A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing* 11, 2 (1997),115—128.
- [4] I. Foster, C. Kesselman, J. M. Nick and S. Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, *presented to OGSF WG, Global Grid Forum*, June 22, 2002.
- [5] Sujata Banerjee, Sujoy Basu, Shishir Garg, Sukesh Garg, Sung Ju Lee, Pramila Mullan, Pruneet Sharma, Scalable Grid Service Discovery Based on UDDI, *Proceedings of the 3rd international workshop on Middleware for grid computing* France, 2005.
- [6] A. Clematis, D. D'Agostino, A. Quarati, A. Corana, V. Gianuzzi, A. Merlo, A Distributed Approach for Structured Resource Discovery on Grid, *International Conference on Complex, Intelligent and Software Intensive Systems*, Barcelone, 2008.
- [7] Globus Alliance, *MDS4: The GT4 monitoring and discovery system*, <http://www.Globus.org/toolkit/docs/4.0/info>, 2005.
- [8] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International J. Supercomputer Applications* 15(3), 2001.
- [9] Géral Lévy, “*Algorithme Combinatoire*”, DUNOD, ISBN 2 10 002149 4, 1994.

6.2. Biographie

Marie Héléne Mballo : Etudiante en thèse à l'école doctorale Mathématique-informatique de Dakar (Sénégal) ; domaine de recherche : découverte de services dans les systèmes P2P.

Ibrahima Niang : Dr. en informatique, enseignant chercheur au dept de Mathématique-Informatique à l'université Cheikh Anta Diop de Dakar (Sénégal). Domaine de recherche : QoS, Sécurité et Mobilité dans le réseau P2P et sans fil.

Yahya Slimani : Professeur en informatique à l'université El Manar Tunis (Tunisie). Domaine de recherche : datamining et fouille de données dans les réseaux P2P et grille de calcul.