

THESE DE DOCTORAT « Programme GET-FET »

Titre : Vers un environnement pour le déploiement logiciel autonome

RESPONSABLE(S) DE LA THESE : Guy Bernard - S. Leriche (recruté au dpt. INF/TMSP en 2007)

La thèse serait en co-encadrement

DOCTORANT :

Profil : Informatique répartie, intergiciel, génie logiciel

Dispositif de financement : CDD

EQUIPE(S) D'ACCUEIL OU SE DEROULERA LA THESE

Equipe « Systèmes répartis » 100%

PROJET (S) STRUCTURANT (S) DE RATTACHEMENT

Projet MARGE

MOTS CLES

Intergiciel, déploiement logiciel, gestion autonome, auto-déploiement, auto-reconfiguration

RESUME

Le déploiement « autonome » de logiciel dans les systèmes répartis à grande échelle (grilles, systèmes pervasifs, P2P...) est une problématique actuelle ouverte. Quelques solutions existent aujourd'hui, mais ne sont exploitables que dans le cadre d'architectures figées. De ce fait, elles ne prennent pas en compte la réalité des évolutions dynamique (variation de QoS, pannes, apparition et disparition de nœuds, de services...) des environnements ouverts.

Nous proposons au cours de cette thèse d'expérimenter une nouvelle approche, basée sur nos précédents résultats dans le domaine des agents mobiles adaptables (AMA). Les propriétés de ces agents logiciels (autonomie, proactivité, adaptation dynamique au contexte...) nous permettent d'envisager la construction d'un intergiciel capable de déployer des applications dans les environnements les plus instables. L'objectif serait de proposer une suite complète de déploiement autonome exploitant cet intergiciel.

Contexte scientifique

1.1 Motivations

Le déploiement d'un système à grande échelle est un processus complexe qui inclut l'installation, la configuration et l'activation de logiciels divers allant du système d'exploitation aux applications utilisateur. Il implique la gestion d'un très grand nombre de machines, de liens de réseau hétérogènes et d'une multitude de versions logicielles. La gestion de tous ces aspects nécessite des outils adaptés qui permettent le contrôle et l'automatisation du processus de déploiement.

Aujourd'hui, pour déployer de manière satisfaisante une application répartie, il est impératif de privilégier une intervention humaine. En effet, les logiciels capables de réaliser un déploiement ne permettent pas de réagir de manière dynamique aux variations de l'environnement (panne de machine, variations de qualité de service...).

Un exemple d'actualité : sur la grille expérimentale GRID5000¹, les chercheurs qui programment un déploiement sur un certain nombre de machines peuvent se retrouver avec moins de ressources que prévues, si les machines tombent en panne entre le moment de leur réservation et leur allocation définitive. Grid5000 est une grille relativement stable, cela permet de donner une idée de la difficulté qu'il existe à déployer de manière automatisée une application répartie à grande échelle dans des infrastructures plus complexes et plus instables telles les réseaux pair à pair (P2P) par exemple.

Nous proposons dans cette thèse d'étudier puis de réaliser un intergiciel qui soit capable de répondre à cette problématique, c'est à dire de déployer « au mieux » une application répartie quel que soit le contexte d'exécution. Par « au mieux », nous entendons que cet intergiciel effectue de manière autonome un déploiement qui satisfasse un ensemble de contraintes spécifiées par l'administrateur en charge du processus de déploiement.

Notre premier cas d'utilisation est le déploiement de COSMOS [3], un intergiciel dédié à l'observation du contexte d'exécution et actuellement développé dans l'équipe MARGE. Pour son utilisation dans l'arc INRIA BROCCOLI², nous envisageons de déployer COSMOS via FDF [6], un intergiciel de déploiement utilisable dans des environnements stables. Toutefois, COSMOS étant prévu pour être utilisé dans des contextes particulièrement instables (réseaux ad-hoc, P2P, systèmes pervasifs...), il nous faut envisager un moyen de le déployer dans ce contexte là, ce que ne peut faire FDF ni aucun autre système de déploiement à l'heure actuelle. Bien sûr, ce n'est qu'un cas d'utilisation, l'intergiciel que nous imaginons étant destiné à déployer n'importe quel logiciel réparti.

1.2 Résumé des objectifs

Dans cette thèse, nous souhaitons dans un premier temps étudier et évaluer l'apport des technologies d'agents mobiles adaptables (AMA) [11] au déploiement de logiciel dans des environnements instables et ouverts tels les systèmes pervasifs ou les réseaux P2P. Puis, dans un second temps, nous voulons réaliser un prototype d'intergiciel de déploiement exploitable dans ce contexte d'exécution sans intervention humaine.

Ainsi, cette thèse permettra de lever le verrou technologique du déploiement logiciel en contexte instable et ouvert.

1.3 Impact

L'impact de cette thèse est attendu sur 3 niveaux :

- Les travaux préliminaires permettront de renforcer la pertinence du modèle d'AMA, et de montrer un nouveau cadre d'expérimentation des technologies *agent logiciel*. Nous pourrions ainsi

¹<http://www.grid5000.fr>

²<http://golgoth.inria.fr/wiki/Contrats/Broccoli>

développer la visibilité de TMSP dans les domaines de l'*autonomic computing* [6], des agents logiciels, et monter de nouveaux partenariats.

- La majorité des projets qui s'adressent aux environnements pervasifs ou à grande échelle font l'hypothèse de l'existence d'un moyen de déploiement, ou encore supposent en pré-requis le bon déploiement du système. Alors qu'il n'existe encore aucune solution logicielle satisfaisante, la réalisation d'un intergiciel de déploiement autonome serait une opportunité d'afficher un leadership sur un domaine en plein essor et de renforcer la position de l'Institut TELECOM en ce qui concerne les technologies et services des réseaux du futur.
- A plus long terme, le prototype de déploiement pourrait être repris et enrichi dans une optique commerciale.

1.4 Innovation et vision à long terme

Actuellement le déploiement d'applications réparties est réalisé principalement à la main, par intervention d'un administrateur qui installe, configure, effectue les mises à jours, etc. de l'application distribuée. Quelques outils permettent de simplifier et d'automatiser le déploiement sous certaines conditions (stabilité de l'architecture sous-jacente, uniformité des composants, absence de pannes), cf. l'état de l'art dans la section suivante. Dans un environnement ubiquitaire, ces différentes actions sont rendues difficiles par l'absence d'hypothèses sur les spécificités ou la disponibilité du matériel.

Nous souhaitons réaliser un intergiciel novateur, qui pourrait à terme être intégré dans une suite complète permettant le *déploiement autonome*, c'est à dire sans intervention humaine et capable de résoudre automatiquement les problèmes liés à l'instabilité et à l'ouverture de l'environnement.

L'innovation réside dans l'utilisation de la technologie « agent logiciel adaptable » pour permettre une adaptation dynamiquement du processus de déploiement.

Un *agent logiciel* est une entité autonome (c'est-à-dire une fonction et les moyens nécessaires pour la calculer) capable de communiquer, disposant de connaissances et d'un comportement privés ainsi que d'une capacité d'exécution propre. Un agent logiciel est dit *adaptable* si certains de ses mécanismes internes, opérationnels (envoi de messages, déplacement. . .) ou fonctionnels (comportement), sont modifiables en cours d'exécution. Conformément à la propriété d'autonomie, l'agent contrôle lui-même ses propres évolutions. Enfin un agent *mobile* peut se déplacer de manière proactive sur le réseau, en transportant ses données, son code et son état d'exécution.

L'utilisation d'agents conduit à la décentralisation de la connaissance et du contrôle. Nous considérons que, par nature, ils constituent un outil privilégié pour l'adaptation et le déploiement en environnement réparti (voir plus de détails dans [11] par exemple).

Ainsi, grâce à leur autonomie, leur proactivité et leurs facultés d'adaptation dynamiques, les agents mobiles adaptables serviront de support au déploiement autonome du logiciel.

1.5 Etat de l'art et compétition

La problématique du déploiement automatisé n'est pas nouvelle, mais les évolutions du contexte vers des environnements instables et ouverts rendent les solutions existantes inefficaces ou carrément inexploitable. Nous présentons ci-dessous les principaux travaux qui inspirent notre vision du déploiement autonome :

SoftwareDock [8] est un prototype universitaire pour le déploiement et la configuration de logiciels dans des environnements répartis. Il permet de gérer intégralement le cycle de vie du déploiement d'un logiciel (installation, activation, mise à jour...), de manière complètement décentralisée. L'architecture de SoftwareDock contient deux parties principales : le *Release Dock* est un serveur placé du côté du producteur de logiciel, contenant les composants applicatifs et pilotant leur déploiement. Chaque site client héberge un *Field Dock*, un serveur qui permet au client de connaître les différentes évolutions possibles (mises à jour, nouveaux logiciels...) et de fournir au Release Dock les informations de configuration du système. Les interactions entre ces éléments ainsi que la mise en oeuvre de toutes les phases du cycle de vie du déploiement est réalisée au moyen d'agents mobiles. Ceux-ci

transportent les composants logiciels sur le réseau, et interprètent leurs descripteurs de déploiement pour réaliser les opérations nécessaires.

L'utilisation d'agents mobiles a donc déjà été proposée, mais pour un contexte fermé (aucune évolution possible de l'architecture), dans lequel les agents ne sont que les vecteurs de déploiement et ne participent pas d'eux même à l'opération de déploiement.

ProActive [1] est un intergiciel de grille supporté par le consortium ObjectWeb. Le modèle de programmation (objets actifs, très proche de nos modèles d'agents logiciels) est adapté au développement d'application réparties à grande échelle. L'outil offre des possibilités de déploiement mais réservées exclusivement aux applications utilisant la bibliothèque ProActive. Il n'est pas possible de déployer des composants ou services d'un autre modèle. Pour répondre aux variations des conditions d'exécution, ProActive propose des moyens d'adaptation avant tout statiques (configuration des protocoles de communication par exemple). L'outil graphique *IC2D* (Interactive Control and Debugging of Distribution) permet d'administrer dynamiquement le système déployé : migration de composants pendant l'exécution, création de nouvelles JVM, etc.

L'approche proposée dans ProActive est très intéressante dans la mesure où elle couvre la quasi totalité du cycle de vie du déploiement. L'outillage accompagnant cette approche permet de : développer, assembler, déployer automatiquement aussi bien les composants que leur support d'exécution (*i.e.* les JVM) sur des machines distantes. Cela reste malgré tout entièrement manuel et donc difficilement gérable, voire impossible dans des environnements à grande échelle ou trop dynamiques.

Deladas (Declarative Language for Describing Autonomic Systems) [5] est un canevas permettant le déploiement d'applications à base de composants distribués. La description du déploiement se fait par l'intermédiaire d'un langage dédié exprimant des contraintes et des relations entre composants. Un solveur de contraintes permet d'obtenir une configuration concrète qui satisfait la description initiale, et d'activer le déploiement. En cas de panne ou de conflit détecté lors de la phase de déploiement, le gestionnaire de déploiement (ADME) peut relancer la phase de résolution de contraintes pour obtenir un nouveau plan de déploiement.

Le langage dédié de Deladas permet au responsable du déploiement d'exprimer simplement ses contraintes, qui seront respectées au mieux par le système de déploiement. Le point faible de cette proposition réside dans le gestionnaire de déploiement (ADME), centralisé, et de la nécessaire réorganisation totale du déploiement à chaque détection d'imprévu. De plus, les auteurs ne présentent pas le fonctionnement d'ADME. Notre proposition de thèse s'intéresse d'abord au système de déploiement, qui ne peut avoir un fonctionnement centralisé pour pouvoir s'exécuter dans un environnement pervasif. Toutefois, dans la vision à long terme d'un canevas complet pour le déploiement autonome, nous pensons nous inspirer du langage de Deladas pour l'expression des besoins de déploiement.

FDF/Deployware (Fractal Deployment Framework) [6] est un canevas de déploiement générique. Ce travail de recherche récent abstrait chaque notion du processus de déploiement sous la forme d'un composant et fournit une manière d'exécuter le déploiement (plan de déploiement). FDF est basé sur le modèle de composants Fractal et son environnement d'exécution. Le processus de déploiement peut être spécialisé en étendant ces composants abstraits nommés *Personnalités*. Une personnalité implémente toute les fonctions nécessaires à l'installation, le démarrage, l'arrêt et la désinstallation de l'application. Cette solution permet d'aboutir à une abstraction de haut niveau cachant au développeur toute la complexité de l'orchestration du déploiement prise en charge par le canevas.

FDF est une solution de déploiement bien outillée pour la validation du déploiement et son exécution, mais ne permet pas de s'adresser aux environnements instables ou ouverts.

2 Contenu scientifique

2.1 Contenu détaillé

Le déroulement de cette thèse est envisagé en trois grandes parties : étude de l'état de l'art du déploiement adaptatif, des environnements instables et des technologies intergicielles ; proposition d'un

modèle de déploiement, conception des architectures logicielles et des outils de génie logiciel associés, réalisation d'un prototype d'environnement de déploiement.

2.2 Appui sur un savoir-faire de l'équipe

Equipe « Systèmes Répartis » / MARGE

Les activités de recherche de l'équipe Systèmes Répartis³ s'effectuent au sein du projet Institut TELECOM « MARGE » (Middleware pour Applications Réparties avec Gestion de l'Environnement). Les objectifs du projet sont de concevoir, réaliser et évaluer des outils pour la construction d'intergiciels à base d'objets ou de composants destinés à supporter l'exécution d'applications distribuées dans des environnements de taille quelconque, impliquant des équipements de nature diverse (y compris des terminaux mobiles) et des utilisateurs éventuellement nomades.

Les travaux s'appuient sur l'expérience acquise par les membres de l'équipe depuis plusieurs années dans les domaines des systèmes d'exploitation répartis à base de micronoyau configurable, du support système pour les applications multimédias, des systèmes répartis à objets, des systèmes à grande échelle, et du partage de charge.

L'expertise de l'équipe dans le domaine des systèmes adaptables et des intergiciels sera un atout pour le déroulement de cette thèse.

3 Conformité de la proposition aux critères de sélection

3.1 Aspect scientifique

Nous présentons cette thèse dans le thème **Réseaux du Futur** des orientations scientifiques et stratégiques de l'Institut TELECOM. En effet, le doctorant devra utiliser des technologies avancées pour les réseaux du futur (composants), le sujet vise un contexte pervasif et instable (ubiquité, transparence, mobilité généralisée). Enfin le déploiement autonome recouvre des activités de gestion autonome des réseaux du futur, au niveau logiciel.

3.2 Encadrement

Sébastien Leriche a été recruté en septembre 2007 dans l'équipe Systèmes Répartis de TMSP pour renforcer le potentiel de l'équipe dans le domaine des intergiciels hautement dynamiques et adaptables au contexte d'exécution. Ses derniers travaux portent sur la construction de modèles d'agents adaptables [11] et de l'exploitation des agents logiciels pour la construction d'applications réparties à grande échelle [12].

3.3 Justification du financement

Ce sujet de thèse est une proposition d'un jeune chercheur pour lever un important verrou technologique avec une approche nouvelle et utilisant des technologies innovantes, pour lequel un financement autre que celui de l'Institut TELECOM est assez improbable.

4 Bibliographie

[1] F. Baude, L. Baduel, D. Caromel, A. Contes, F. Huet, M. Morel and R. Quilici : *Programming, Composing, Deploying for the Grid*. In GRID COMPUTING: Software Environments and Tools, Jose C. Cunha and Omer F. Rana (Eds), Springer Verlag, January 2006.

[2] D. Conan, R. Rouvoy, L. Seinturier: *COSMOS: composition de noeuds de contexte*. Technique et Science Informatiques, à paraître en 2008.

³http://www.it-sudparis.eu/inf/recherche/systemes_repartis.php

- [3] D. Conan, R. Rouvoy, and L. Seinturier : *Scalable Processing of Context Information with COSMOS*. Proc. 7th IFIP International Conference on Distributed Applications and Interoperable Systems, Springer-Verlag, Lecture Notes in Computer Science Volume 4531, pages 210-224, Paphos, Cyprus, June 2007.
- [4] J. Coutaz, J. Crowley, S. Dobson, and D. Garlan. *The disappearing computer : Context is Key*. Communications of the ACM, 48(3) :49–53, Mar. 2005.
- [5] A. Dearle, G.N.C. Kirby, A.J. McCarthy : *A framework for constraint-based development and autonomic management of distributed applications*. Proc. Intl. Conf. On Autonomic Computing 2004, May 2004.
- [6] A. Flissi, J. Dubus, N. Dolet, P. Merle : *Deploying on the Grid with DeployWare*. ccgrid, pp. 177-184, 2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID), 2008.
- [7] D. Hoareau, T. Abdellatif, Y. Mahéo : *Architecture-Based Autonomic Deployment of J2EE Systems in Grids*. In International Conference on Grid and Pervasive Computing (GPC'07), LNCS, Paris, France, May 2007.
- [8] R. Hall, D. Heimbigner, A. van der Hoek et A. Wolf : *The software dock : A distributed, agent-based software deployment system*. Rapport technique CU-CS-832-97, Department of Computer Science, University of Colorado, 1997.
- [9] M.C. Huebscher, J.A. McCann : *A Survey of Autonomic Computing – Degrees, Models and Applications*. ACM Computing surveys, Vol. 40, No. 3, Article 7, August 2008.
- [10] K.O Kehpart, D.M. Chess : *The vision of autonomic computing*. Computer 36, 1, 41–50, 2003.
- [11] S. Leriche, J.-P. Arcangeli : Agent[®]: A Tool for Modeling Composite Self-Adaptive Agents. International Transactions on Systems Science and Applications, Vol. 4, No. 2, May 2008.
- [12] Sébastien Leriche, Jean-Paul Arcangeli, Marc Pantel : *Un framework à composants et agents pour les applications réparties à grande échelle*. Dans : Revue des Sciences et Technologies de l'Information, série L'Objet, Lavoisier, Vol. 12, N. 4, p. 103-132, décembre 2006