

Distribution d'un prouveur SAT

Maîtrise Informatique 2003-2004

Frédéric Fontaine

Sujet proposé par Mr Le Berre
(leberre@cril.univ-artois.fr)

Le sujet

- **Objectifs :**
 - Apprentissage de ProActive.
 - Distribution du prouveur MINISAT réalisé en Génie Logiciel lors du premier semestre.
 - Distribution de la bibliothèque Sat4J.
- **Raison : proposition de l'équipe ProActive**
 - Distribution d'un prouveur Sat (en java) à l'aide de ProActive.
 - Prouveur open source.



ObjectWeb et projet Oasis

- **ObjectWeb**

- Communauté pour les logiciels open source créé par Bull, France Telecom et l'INRIA (1999).
- Evolution sous forme d'un consortium international accueilli par l'INRIA (2002).
- Ambition : réunir une collection cohérente de projets.

- **Projet Oasis**

- Projet commun entre le laboratoire CNRS I3S et l'Université de Nice Sophia Antipolis.
- Construire un environnement fondé sur la sémantique (pour le développement d'application réparties et communicantes).
- Construction d'une bibliothèque facilitant la programmation d'application distribuée : ProActive.



ProActive : Fonctionnalités

- Bibliothèque java pour les applications parallèles, distribuées et concurrentielles.
- API détaillée.
- Basé sur les objets actifs.
- Fabriqué à l'aide des classes Java standard.
- Bibliothèque extensible.



Les outils utilisés

- **MiniSat**

- Prouveur SAT minimal, simplification de la conception de SATZOO et SATNIK.
- SATZOO a remporté 2 catégories lors la compétition SAT2003 et 1 en 2004.
- En MI3GL, réalisation d'un prouveur suivant les spécifications de MINISAT.

- **Sat4J**

- Bibliothèque efficace de prouveur SAT en java.
- Inclut les spécifications du prouveurs MiniSat.
- MiniLearning est basé sur SAT4J. Il a été soumis à la compétition SAT2004.
- Utilisé comme base au CRIL pour la recherche sur les formules pseudo booléenne.



Technologies Java utilisées

- Bibliothèques ASM et BCEL : manipulation du bytecode pour la création des objets actifs.
- Sérialisation : autoriser le déplacement des objets.
- XML : fichier de description.
- Technologie RMI, JINI : gestion des objets distants.
- Protocoles : **rsh, rlogin, ssh, lsf, globus.**



Objet actif



Introduction

Outils utilisés

ProActive

Travail réalisé

Problèmes rencontrés

Conclusion

Objet actif

- **Objet standard**



Introduction

Outils utilisés

ProActive

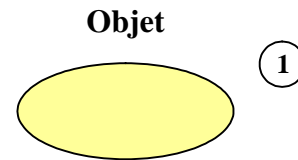
Travail réalisé

Problèmes rencontrés

Conclusion

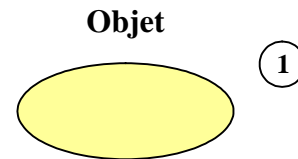
Objet actif

- **Objet standard**



Objet actif

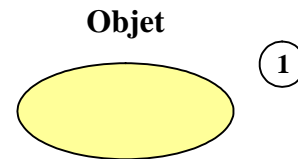
- Objet standard



- Objet actif

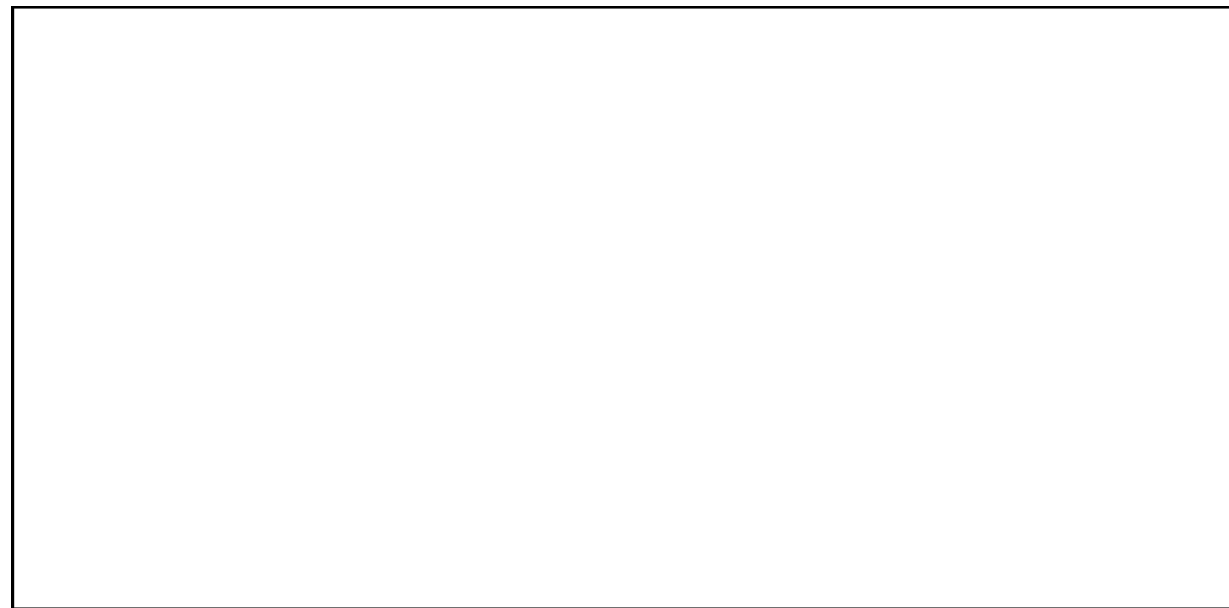
Objet actif

- Objet standard



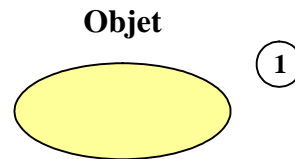
- Objet actif

Objet Actif

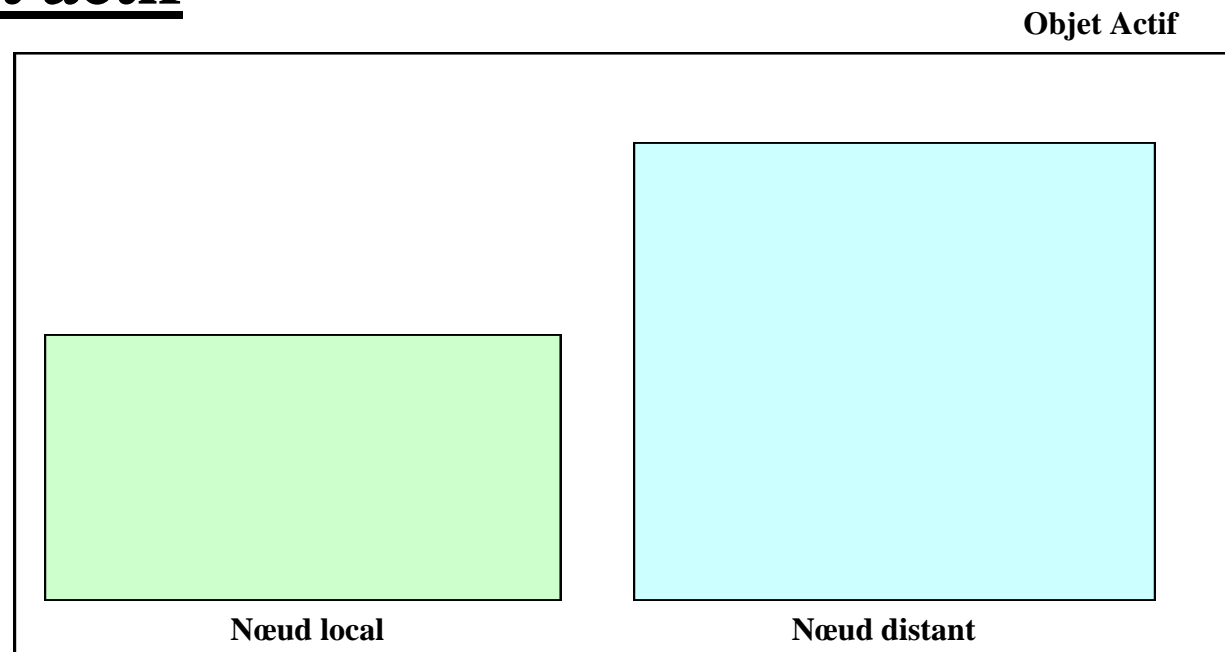


Objet actif

- Objet standard

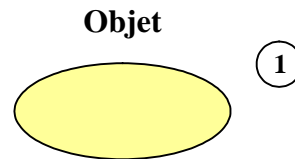


- Objet actif

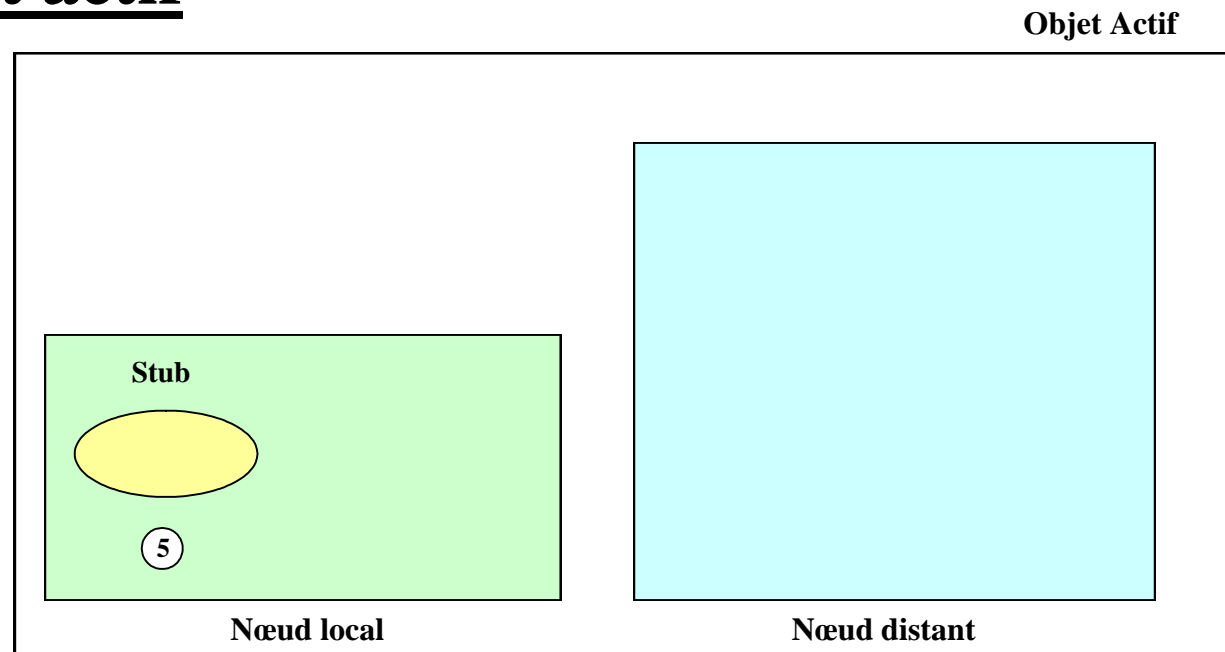


Objet actif

- Objet standard

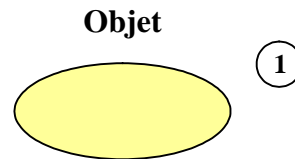


- Objet actif

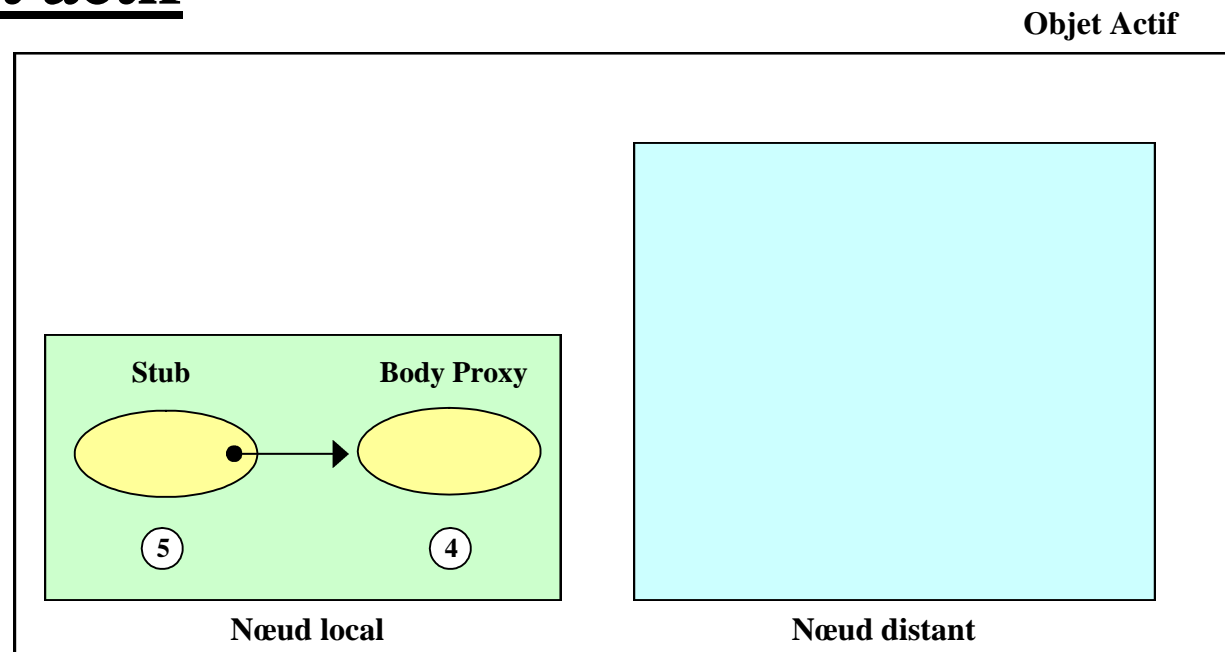


Objet actif

- Objet standard



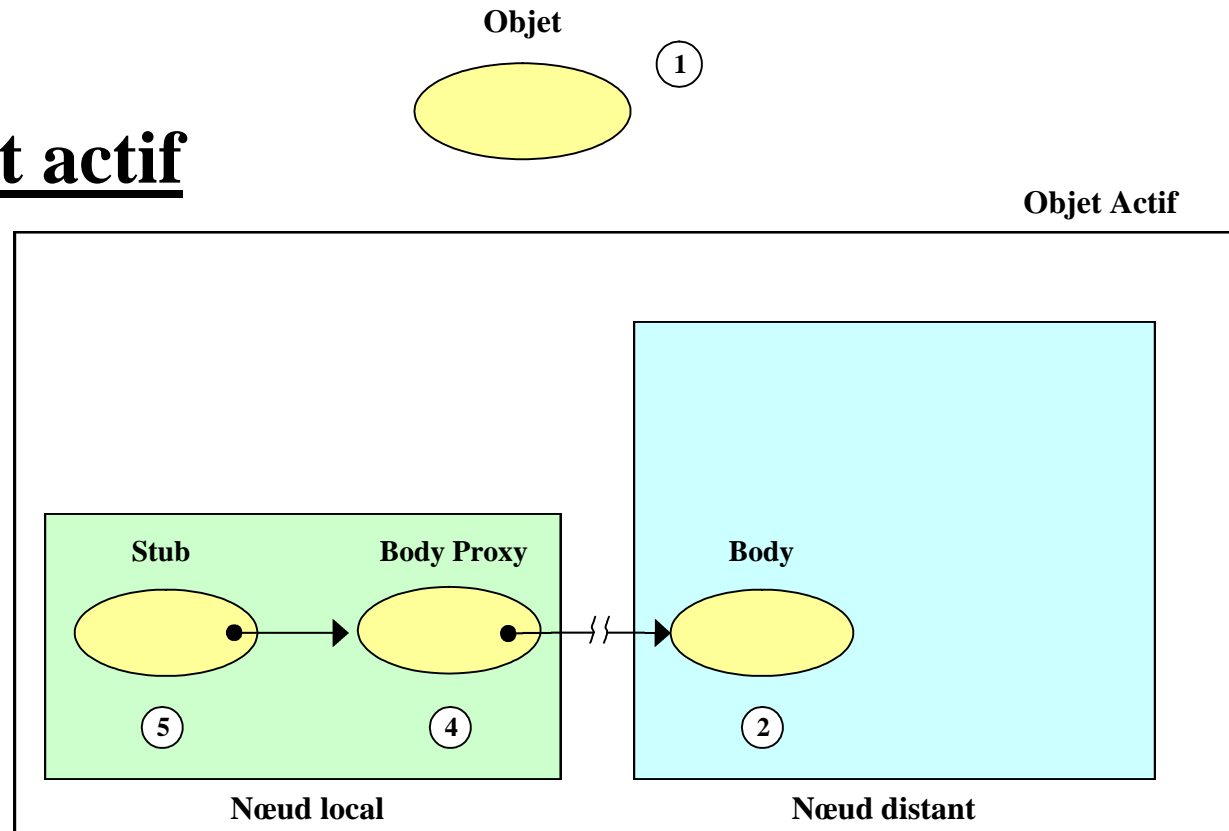
- Objet actif



Objet actif

- Objet standard

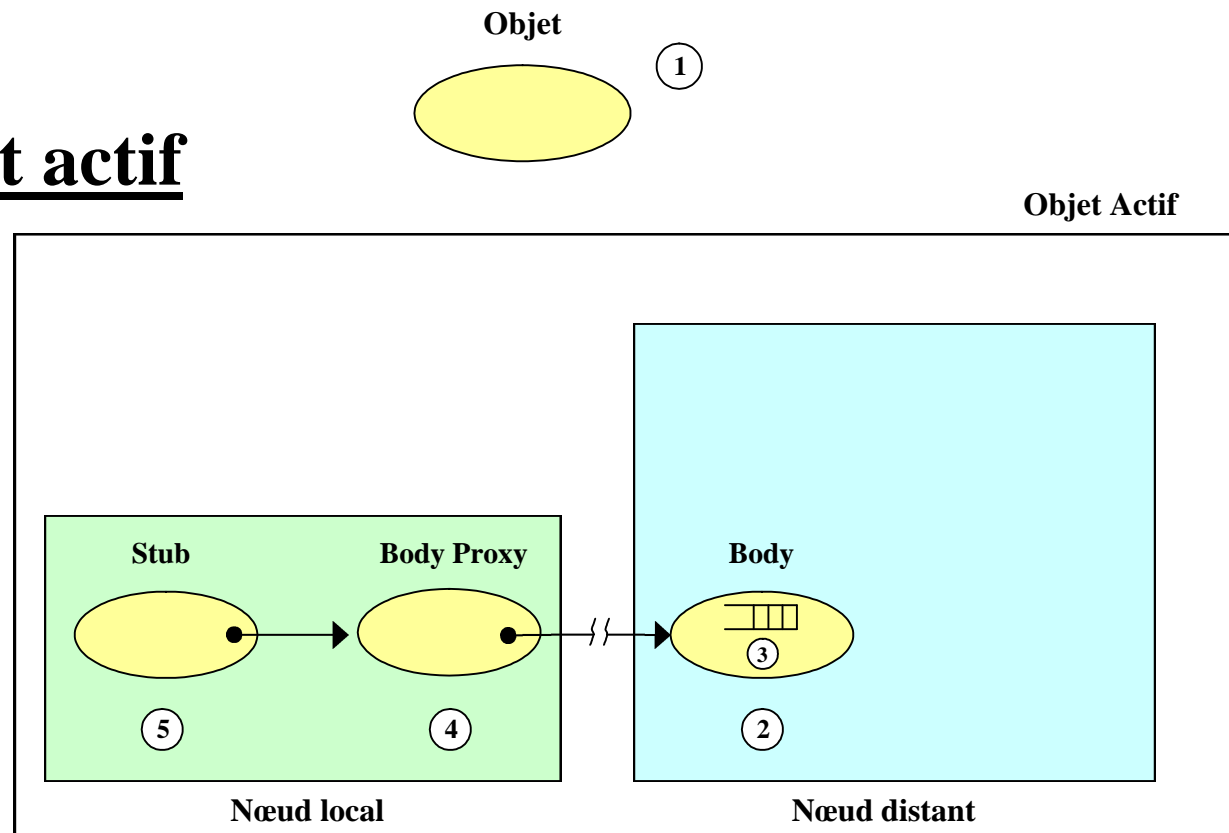
- Objet actif



Objet actif

- Objet standard

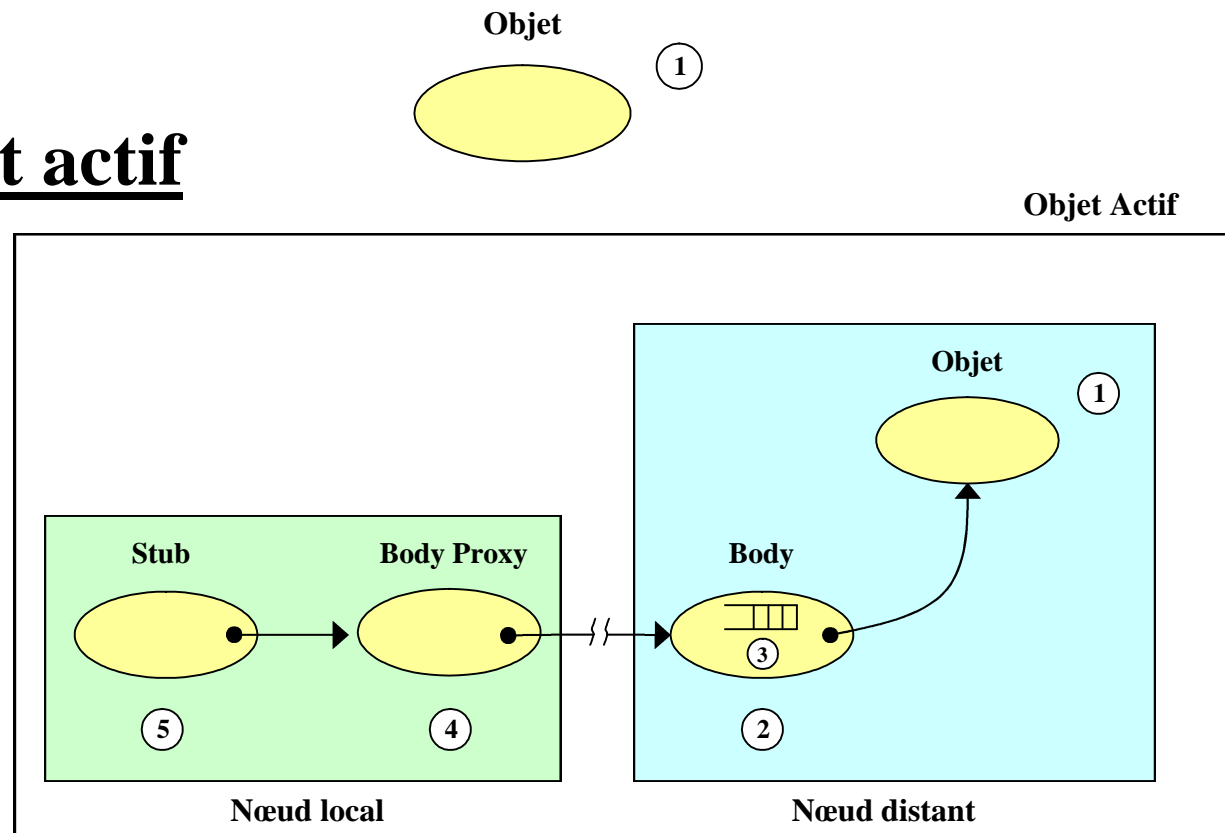
- Objet actif



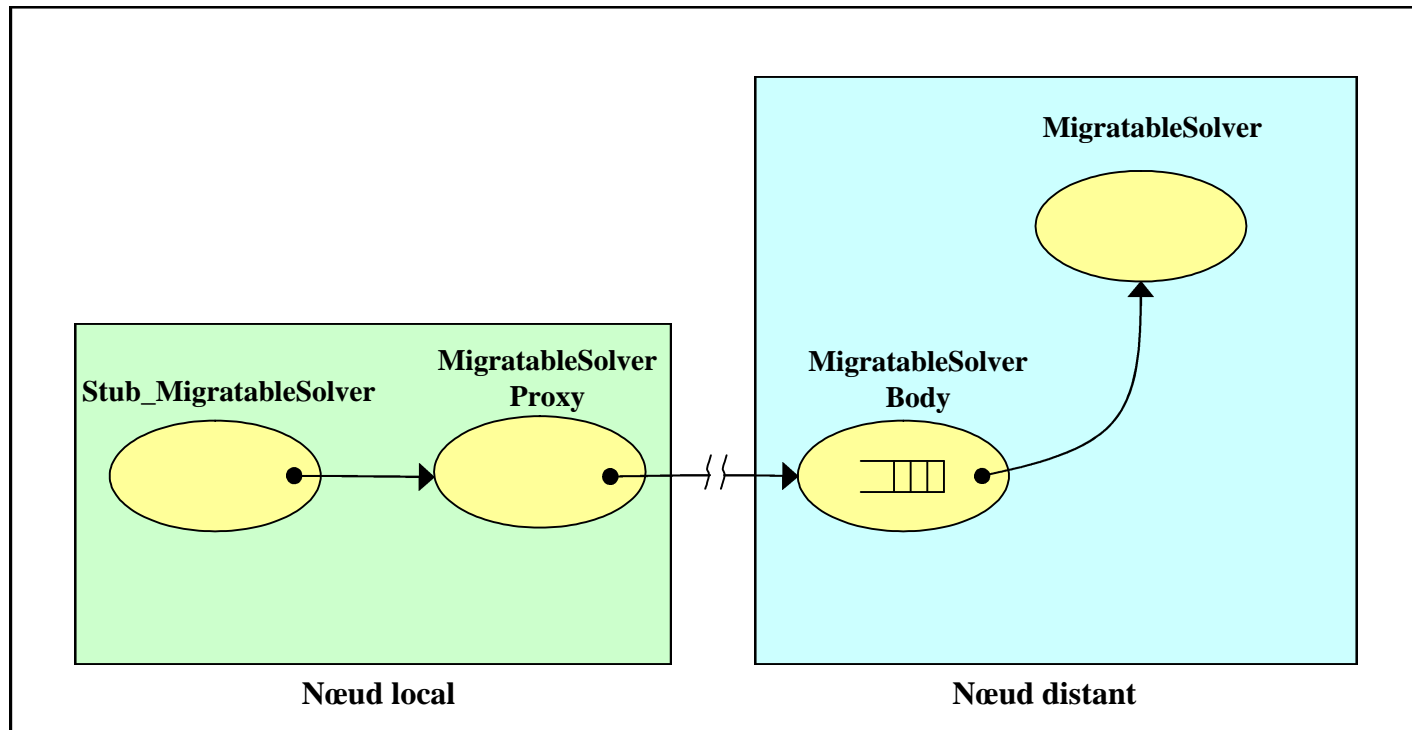
Objet actif

- Objet standard

- Objet actif



Exemple



Travail Réalisé

- **MiniSat**

- Connaissance du code (projetMI3GL).
- But : Comprendre le fonctionnement de ProActive.
- Coder une version distribué du prouveur.

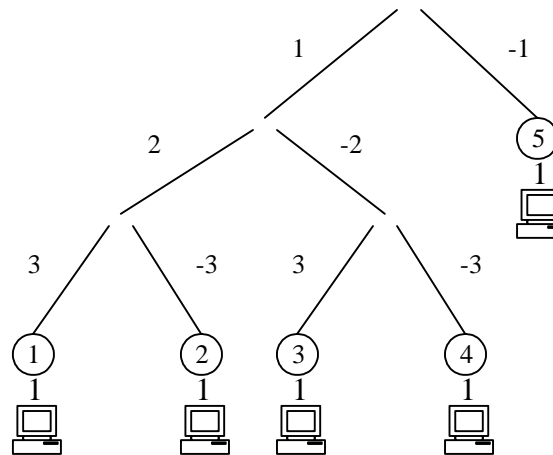
- **Sat4J**

- Analyse du code existant.
- Coder une version distribué du prouveur.
- Minimiser le coût de la distribution.



Distribution de MiniSat

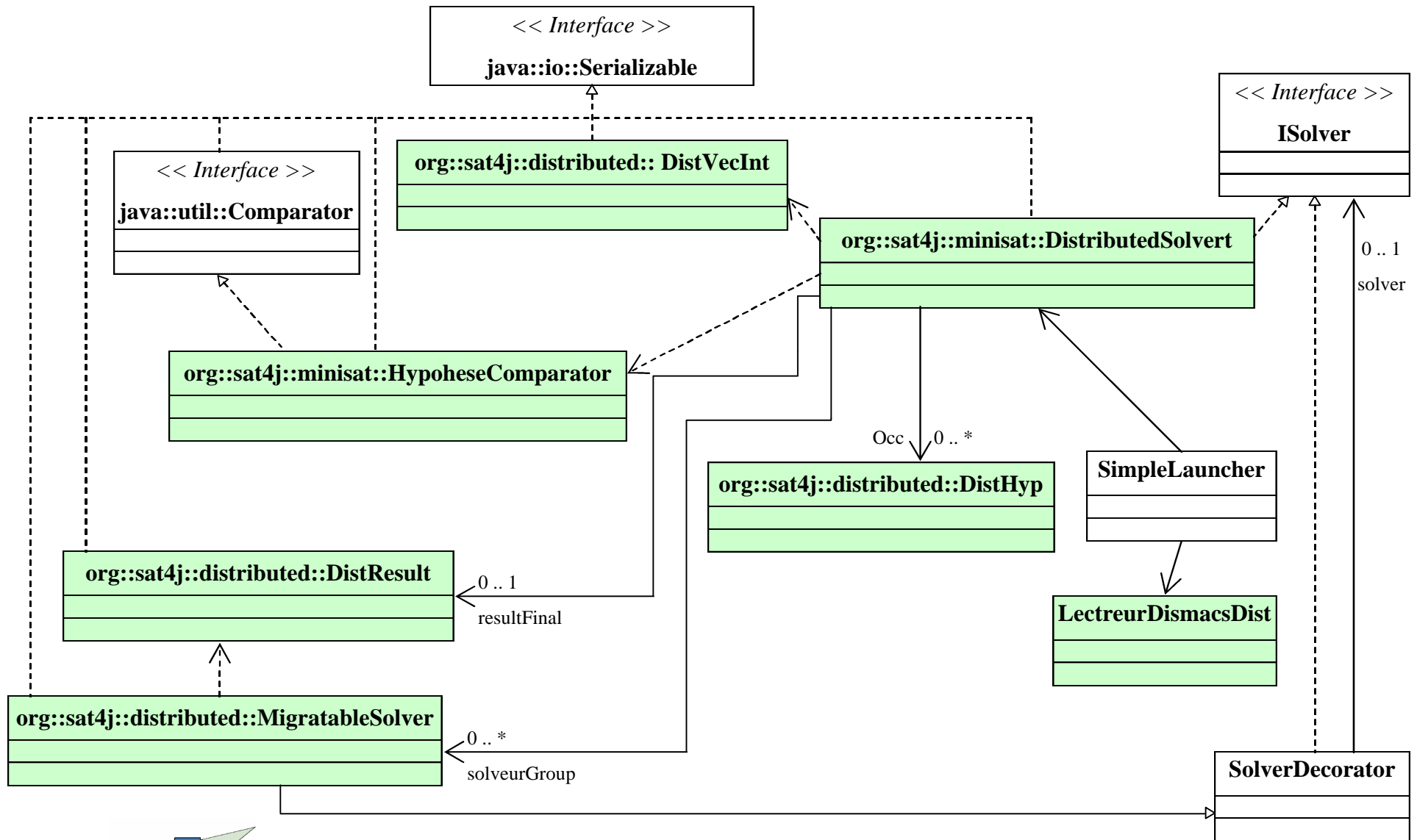
- Compréhension du fonctionnement de ProActive :
 - Fonctionnement des objets actifs.
 - Fonctionnement des groupes.
- Hypothèses différentes fournies au prouveur.



Distribution de Sat4J

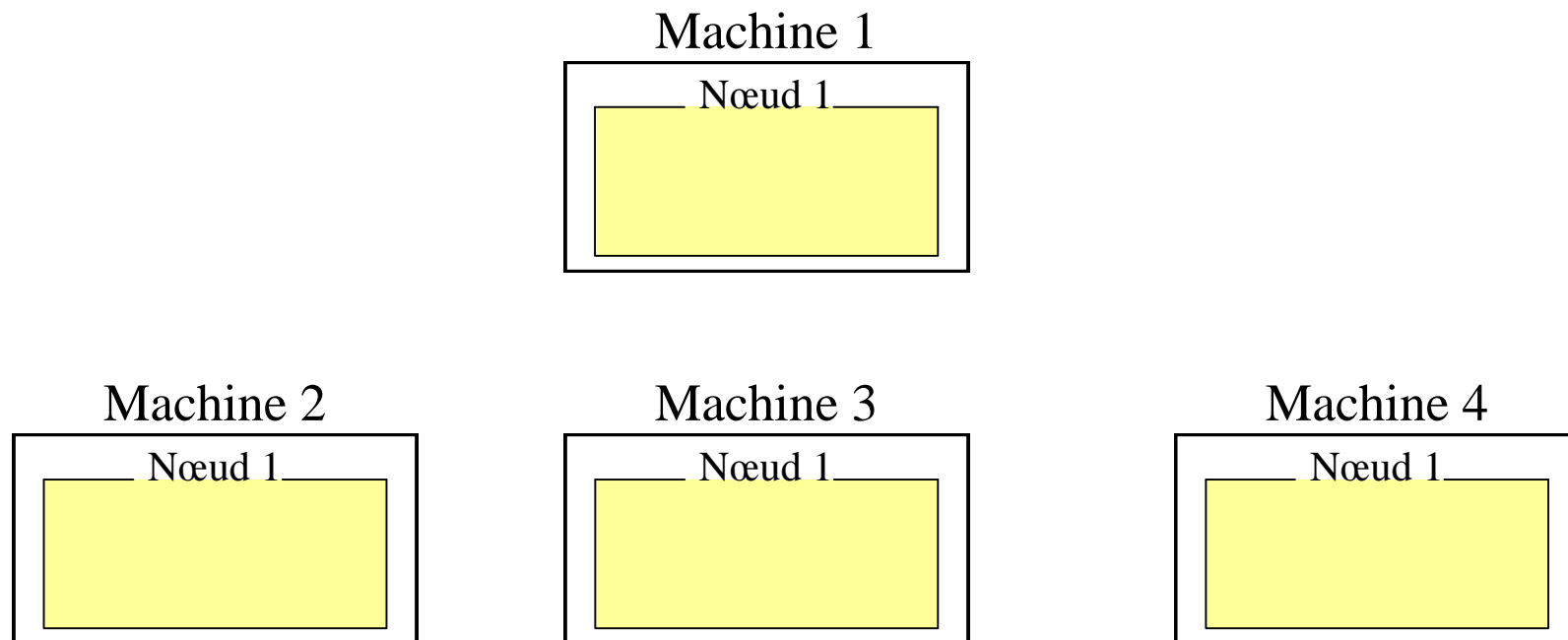
- Utilisation de fichier XML de déploiement
 - Facilité/souplesse d'utilisation
 - Nécessite une installation réseau particulière (ssh+ lsf par exemple)
- Minimiser le code supplémentaire dans SAT4J (miniactive) : seulement 7 nouvelles classes.
- Minimiser le coût de la distribution :
 - Limiter le surcoût de la Sérialisation (lors de la lecture des clauses).
 - Choix des hypothèses les plus apparentes.

Solution proposée



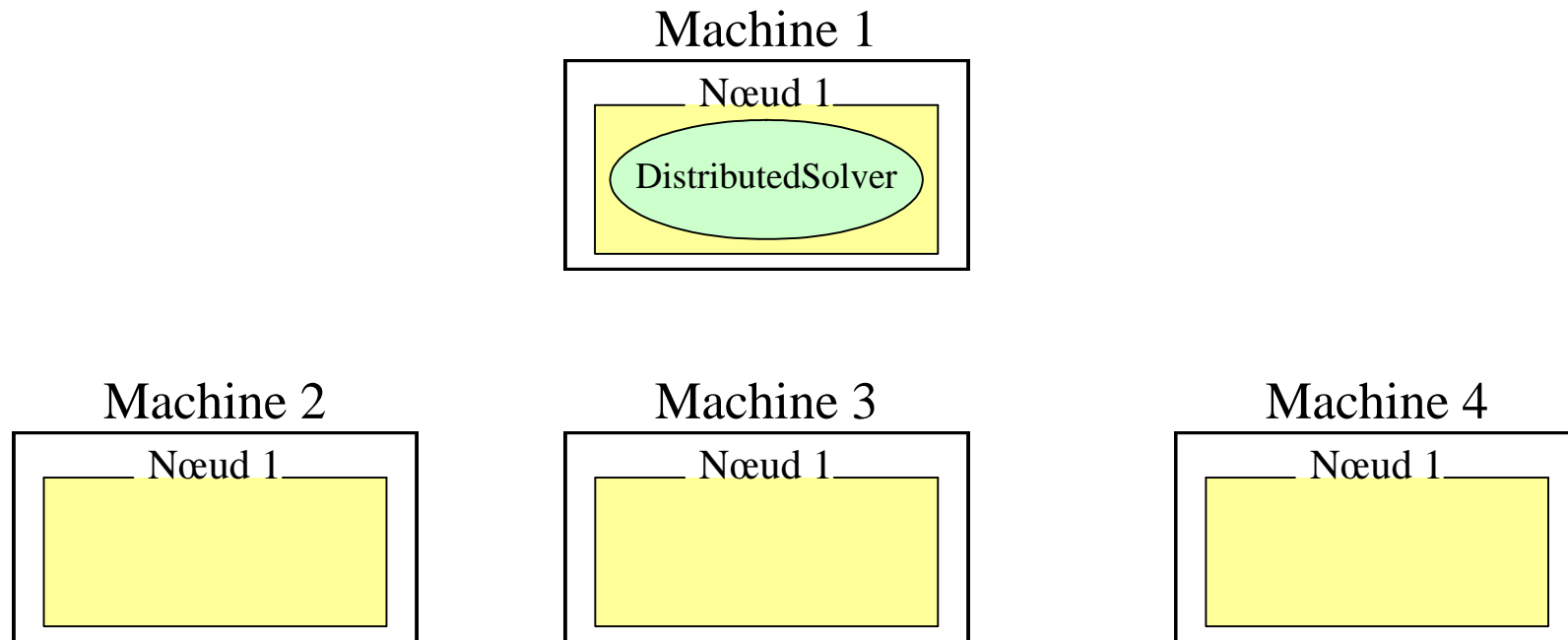
Solution proposée

- **Distribution sur les noeuds**



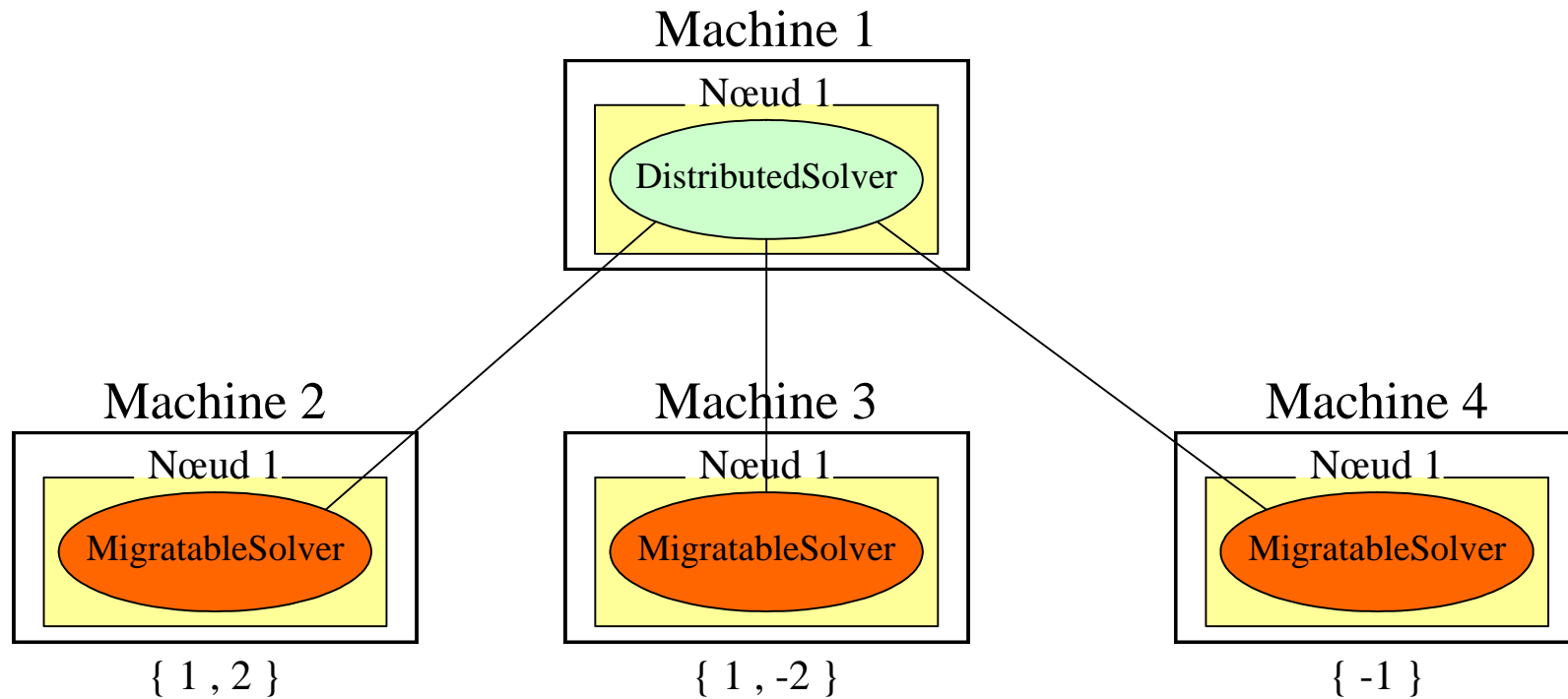
Solution proposée

- **Distribution sur les noeuds**



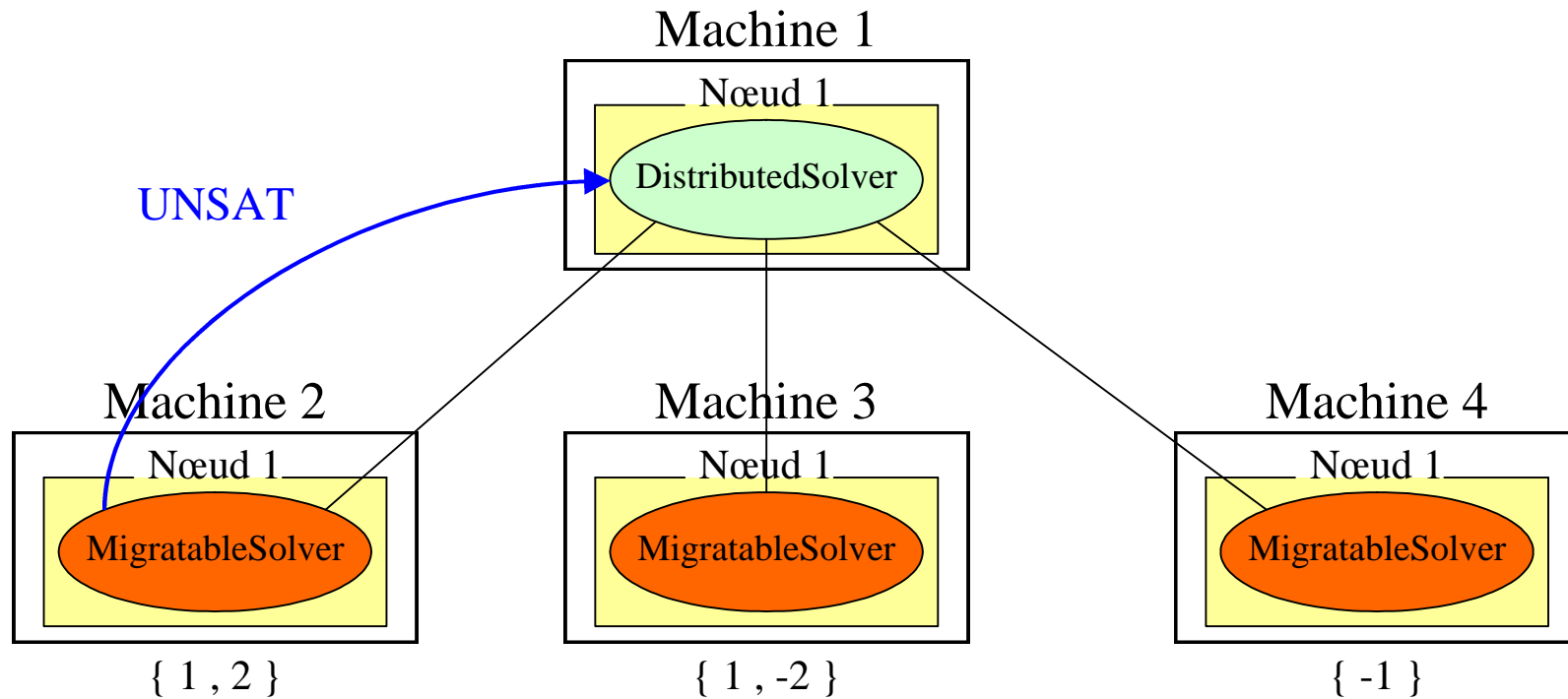
Solution proposée

- Distribution sur les noeuds



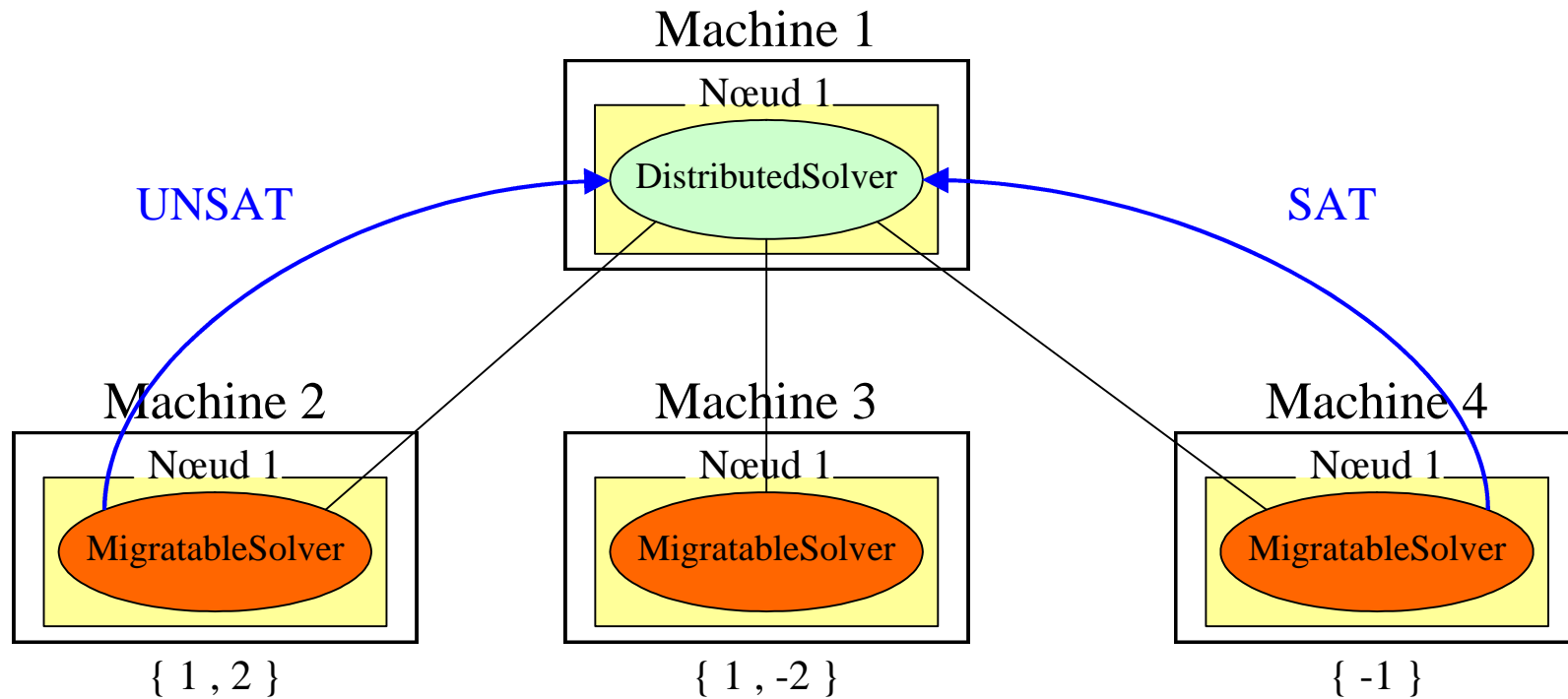
Solution proposée

- Distribution sur les noeuds



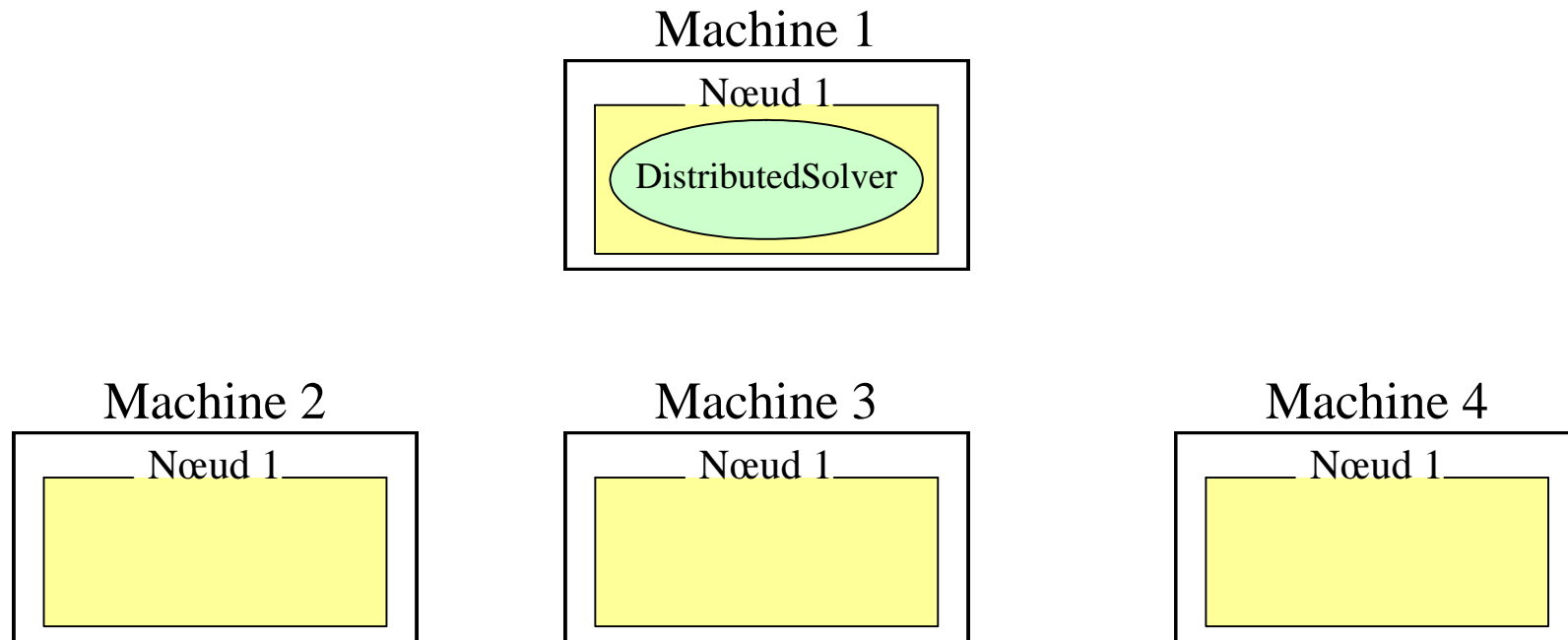
Solution proposée

- Distribution sur les noeuds



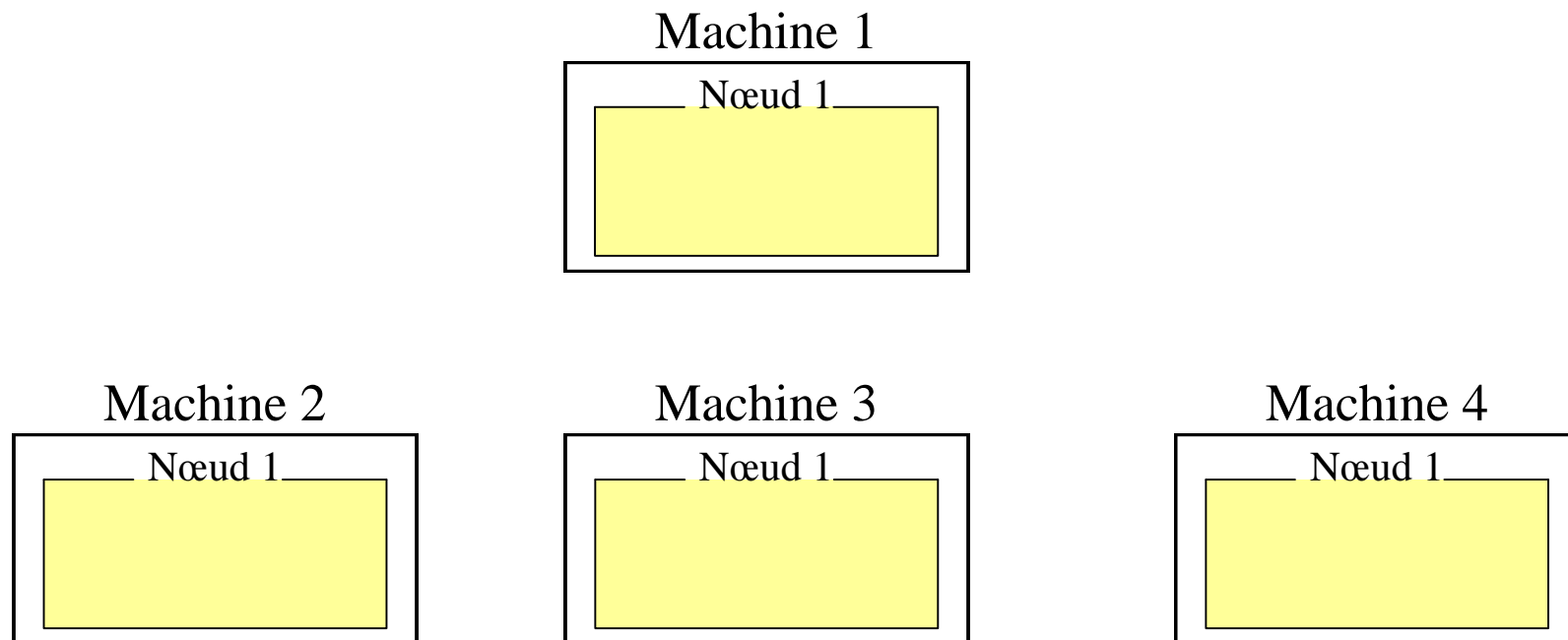
Solution proposée

- **Distribution sur les noeuds**



Solution proposée

- **Distribution sur les noeuds**



Résultats obtenus

- Résolution d'une instance SAT (IBM, #vars 22928, #clauses 116491)

Nombre de noeuds	Temps lecture fichier CNF	Temps total résolution(s)
1	9 s	57 s
2	12 s	90 s
3	15 s	58 s
4	17 s	74 s
5	18 s	48 s
6	20 s	51 s

Démonstration



Introduction

Outils utilisés

ProActive

Travail réalisé

Problèmes rencontrés

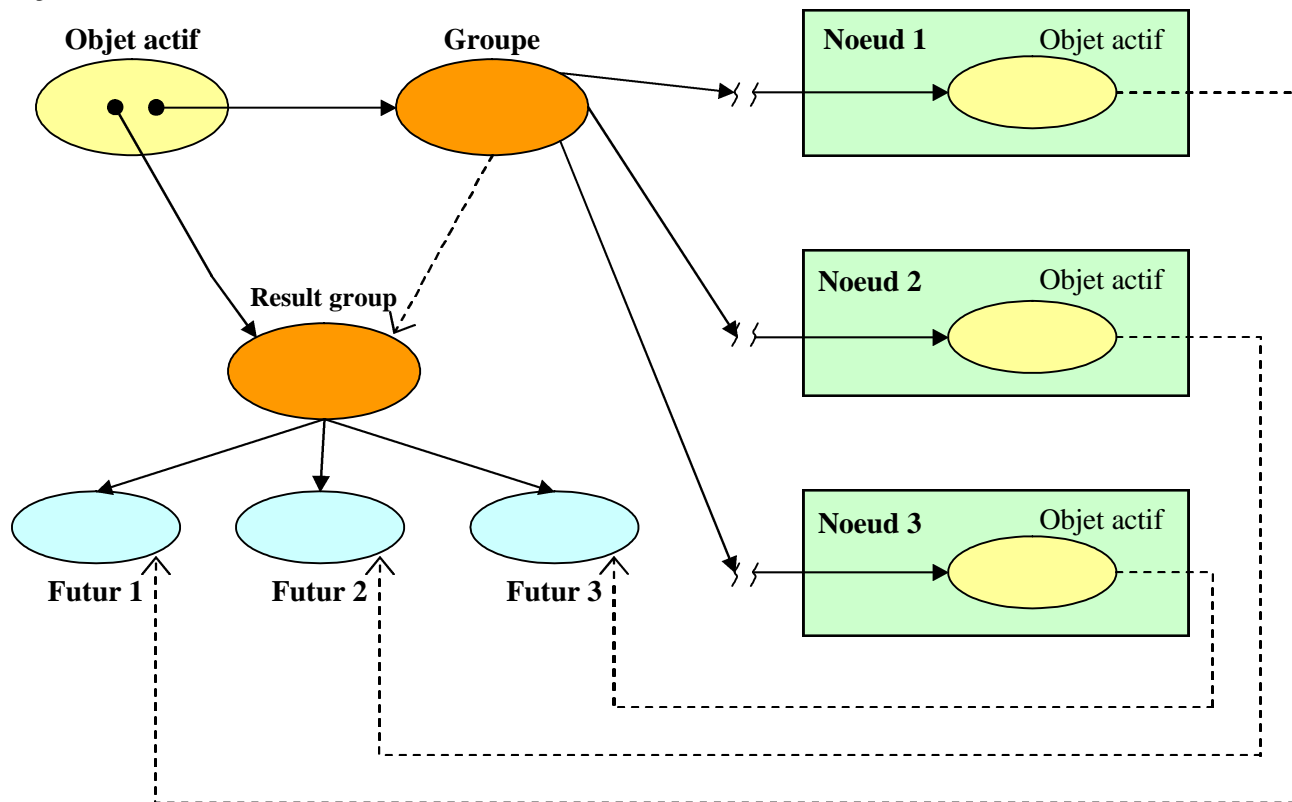
Conclusion

Problèmes rencontrés

- Restriction de ProActive
 - constructeur sans paramètres.
 - retour autre que type primitif (retour d'un objet non final).
 - pas d'appel asynchrone sur une méthode pouvant lancer une exception.

Problèmes rencontrés

- Synchronisation



Conclusion

- premier projet informatique
- approfondir mes connaissances sur java et les systèmes distribués
- découvrir le fonctionnement d'un laboratoire de recherche

