Introduction

Apres avoir etabli des bases solides au projet lors de la premiere partie, nous allons a present perfectionner les fonctionnalites existantes et les etendre. C'est ainsi que nous avons divise notre travail en deux axes distincts. Le premier ayant pour objectif d'ameliorer les performances, le second de rendre l'interface Linda utilisable sur des machines differentes.

Parallelisation des operations Linda

Comme l'ont montre les tests de performance effectues precedemment, Linda est un processus de synchronisation lent. Cette degradation vient avec l'accroissement du nombre d'objet a parcourir lors d'une recherche, la rendant immanquablement plus lente. Pour pallier a ce probleme, nous avons decide de separer l'espace de tuple. Vous trouverez deux versions implantant cette idee.
La premiere ThreadedCentralisedLinda se base sur l'utilisation d'un Tuplespace qui se charge de repartir les tuples dans plusieurs espaces distincts. Le Tuplespace implante l'interface Collection et possede une methode get, ce qui le rend immediatement compatible avec l'implantation de Linda effectue en partie 1. La seconde DecentralisedLinda concoit un Linda comme un regroupement d'autres Linda. Il repartit les operation entre ces differentes instances dans des processus distincts. Le gain de performances est nettement visible sur cette seconde version, mais on constate une degradation forte avec la premiere version. Il faudrait donc investiguer sur ce probleme pour une version future.