Proposition de recherche doctorale

**Garbage collection for managed runtime on multicore processors**

**Mots clés :**
- Directeur de thèse : Marc Shapiro
- Co-encadrant(s) :
- Unité de recherche : Laboratoire d'informatique de Paris 6
- École doctorale : École Doctorale Informatique, Télécommunications, Électronique de Paris
- Domaine scientifique principal : Divers

**Résumé du projet de recherche (Langue 1)**

Managed Runtime Environments (MREs), such as a JVM or a CLI, are increasingly successful for application development and execution, mainly because they provide good properties of manageability, isolation, and safety in general. A key component is the automated memory manager or garbage collector (GC). Garbage collection automatically frees unused memory, relieving application programmers from complex protocols and guaranteeing the absence of memory leaks or access violations. However, this comes at a price. Experimentally, GC may add up to 20% or 30% to mutator (i.e., application) execution time. This performance penalty is particularly an issue with the recent emergence of multicore architectures. Indeed, current GC algorithms used in production MREs environments execute sequentially and block all mutator threads. Amdahl's law predicts that, due to this sequential bottleneck, even hundreds or thousands of cores will not speed up a mutator any better than a factor of five. The performance of mutators cannot scale with the number of cores. Furthermore, the performance impact of GC on multicore architectures is exacerbated by the memory architecture. Indeed, current GC algorithms destroy cache locality, leading to numerous cache misses, and further decreasing performance.

**Informations complémentaires (Langue 1)**

Stage à l'étranger obligatoire pendant la thèse