Proposition de recherche doctorale

Memory Snapshot in a Java Virtual machine for Constraint Solvers

Mots clés :

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

Résumé du projet de recherche (Langue 1)

Scientific context: Pervasive computing is now a reality with the massive deployment of mobile appliances, particularly smart phones. Despite the increasing importance of pervasive applications, providing adequate support for them within a computing environment remains a challenge. Pervasive applications vary greatly in their resource requirements (processors, memory bandwidth, and disks): for critical applications, such as surveillance systems, a known bounded amount of resources must be guaranteed for the application to run properly; for multimedia applications, the amount of resources needed may change over time and dynamic adaption is often required. To manage these differing and possibly varying resource requirements, a system of resource reservations is required. Because the availability of resources is global to a computing environment, such reservations have to be managed at the middleware level. Modern pervasive middleware is typically implemented using Java, for example with OSGi or Android, because of its safety, flexibility, and mature development environment. However, the Java virtual machine specification has not been revised since 1999, at the time when the idea of pervasive computing was first introduced. It was designed to execute only a single application at a time, and thus it does not provide resource accounting or per-application resource reservations. Current pervasive middlewares are thus unable to reserve resources for critical applications, which may cause these applications to crash or hang when insufficient resources are available, and are unable to provide resource accounting, making it impossible to balance the load on the devices and to optimize resource use.

Résumé du projet de recherche (Langue 2)

PhD topic: Globally optimizing resources require complex analyses to balance the load between the devices. These analyses must take into account the various available resources (CPU, bandwidth, memory, disk), the resource consumption of each application, the criticality of applications and the fact that some applications are highly inter-dependent and should be placed on the same device. Constraint solvers are well adapted for finding this optimal placement. Constraint solving requires searching through a state space, which is typically implemented using a backtracking algorithm that induces frequent saving and restoring of memory states. However, intensive memory copy is known to be a bottleneck on modern architectures that have complex cache hierarchies. Furthermore, copying objects in Java can only be implemented at the field level, which prevents the use of well known low-level optimizations. To improve the performance of constraint solver, the PhD student will modify the VMKit Java virtual machine to reuse memory hardware protection provided by the operating system to build copy-on-write segments. A copy-on-write segment is a contiguous memory zone where memory pages are only copied when they are written. Because writes are detected by the Memory Management Unit hardware, the use of a copy-on-write segment avoids the need to build costly "backtrackable" objects and will reduce the cost of copies. Bibliography: N. Geoffray, G. Thomas, J. Lawall, G. Muller, and B. Folliot. VMKit: a Substrate for Managed Runtime Environments. In Proceedings of VEE 2010, pages 51-62, Pittsburgh, USA, Mar. 2010. ACM. CHOCO Team. Choco: an Open Source Java Constraint Programming Library. Research report 10-02-INFO. Ecole des Mines de Nantes. 2010.