

A Distributed Runtime for Java: Yesterday and Today

Michael Factor
IBM Research Lab in Haifa
Haifa University Campus
Haifa 31905, Israel
factor@il.ibm.com

Assaf Schuster
Computer Science Department
Israel Institute of Technology
Technion City
Haifa 32000, Israel
assaf@cs.technion.ac.il

Konstantin Shagin
Computer Science Department
Israel Institute of Technology
Technion City
Haifa 32000, Israel
konst@cs.technion.ac.il

Abstract

Since the introduction of the Java language less than a decade ago, there have been several attempts to create a runtime system for distributed execution of multithreaded Java applications. The goal of these attempts was to gain increased computational power while preserving Java's convenient parallel programming paradigm. This paper gives a detailed overview of the existing distributed runtime systems for Java and presents a new approach, implemented in a system called JavaSplit. Unlike previous works, which either forfeit Java's portability or introduce unconventional programming constructs, JavaSplit is able to execute standard multithreaded Java while preserving portability. JavaSplit works by rewriting the bytecodes of a given parallel application, transforming it into a distributed application that incorporates all the runtime logic. Each runtime node carries out its part of the resulting distributed computation using nothing but its local standard (unmodified) Java Virtual Machine (JVM).